CONTEXT AWARE ROUTING MANAGEMENT ARCHITECTURE
FOR AIRBORNE NETWORKS

THESIS

Joan A. Betances, Captain, USAF

AFIT/GCE/ENG/12-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States

AFIT/GCE/ENG/12-01

CONTEXT AWARE ROUTING MANAGEMENT ARCHITECTURE FOR AIRBORNE NETWORKS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Insitute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Engineering

Joan A. Betances, B.S.E.E., B.S.C.S.

Captain, USAF

March 2012

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GCE/ENG/12-01

CONTEXT AWARE ROUTING MANAGEMENT ARCHITECTURE FOR AIRBORNE NETWORKS

Joan A. Betances, B.S.E.E., B.S.C.S.
Captain, USAF

Approved:

// Signed //	15-March-2012
Dr. Kenneth M. Hopkinson, PhD (Chairman)	Date
// Signed //	15-March-2012
Major Jeffrey M. Hemmes, PhD (Committee Member)	Date
// Signed //	15-March-2012
Major Mark D. Silvius, PhD (Committee Member)	Date

AFIT/GCE/ENG/12-01

Abstract

This thesis advocates the use of Kalman filters in conjunction with network topology information derived from the Air Tasking Order (ATO) during the planning phase for military missions. This approach is the basis for an algorithm that implements network controls that optimize network performance for Mobile Ad hoc Networks (MANET). The trajectories of relevant nodes (airborne platforms) participating in the MANET can be forecasted by parsing key information contained in the ATO. This information is used to develop optimum network routes that can significantly improve MANET performance. Improved MANET performance in the battlefield enables decision makers to access information from several sensors that can summarize mission execution status realtime. In one simulated test case there was a 25% percent improvement of network throughput, and 23% reduction on dropped packets. Using this technique we can selectively preserve the Quality of Service (QoS) by establishing network controls that drops low priority packets when necessary. The algorithm improves the overall MANET throughput while minimizing the packets dropped due to network congestion.

To My Wife and Kids.

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Dr. Hopkinson, for his guidance and support throughout the course of this thesis effort. I would also like to show appreciation to Mr. Kerner for his programming assistance. Additionally, I would also like thank everyone in the Cyber Advanced Networking in Mobile Applications Laboratory (ANiMaL) especially Captain Kasperek, Captain Morales, Captain Simpson, Captain Carbino and Major Fadul for their assistance and support provided to me in this endeavor.

Joan A. Betances

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Context Aware Routing Management Architecture for Airborne Networks

1 Introduction

There is no doubt that in future conflicts the performance of airborne networks used to support military missions will be a critical enabler to achieve dominance in the battlefield. Increased battlespace awareness is achieved as information networks are optimized for access and delivery of critical information. As networks are optimized, key information obtained from sensors in the battlefield can be disseminated to decision makers faster. This concept has enabled the US military to detect and destroy targets within minutes by leveraging automated Intelligence, Surveillance and Reconnaissance (ISR) systems [36].

In spite of the achieved progress in airborne networks, there are several challenges that are inherent in them. Protocols designed to be used with the Internet do not perform well in an exclusively military environment. The main objective for the protocols that are currently used to provide services over the Internet is to deliver electronic data between network nodes reliably by detecting and avoiding faulty network paths. Traditional Internet protocols do not incorporate measures to react quickly enough to dynamic networks where the topology changes frequently. Classical routing algorithms are inefficient in high mobility and scalability scenarios [24] as normally encountered in military operations. Additionally, adversaries might resort to interference or jamming in order to degrade network performance. Existing protocols cannot handle this wide range of challenges.

Much research has been undertaken to modify existing wired and wireless ad hoc network protocols so that they can better support Department of Defense (DoD) needs.

Traditional research for wireless ad hoc networks is focused on addressing the scalability issues by modeling mobile network with thousands of nodes that move randomly. This type of research may lead to novel ideas, but they do not address the issues normally encountered in tactical military operations. Military networks are normally limited to hundreds of nodes.

Traditional network routing algorithms were designed with a fixed probability that a given failure would occur. Based on that fixed probability for expected failure, mechanisms are incorporated to provide optimum performance under this specific circumstances. However, this stochastic model does not hold true in a dynamic environment [28].

The mobility of nodes in a given tactical airborne scenario is driven by requirements of the military mission. These preplanned requirements are documented in the Air Tasking Order (ATO) [15]. During the ATO generation phase, it is possible to gather information about the trajectories of the different airborne platforms participating. Additionally, specifics about the type of network support required in order to execute the mission can also be obtained. By taking into consideration the capabilities of each participating node and how they are used it is possible to generate efficient network routes optimizing the performance of the overall network. Additionally, this information can be used to prioritize packets and generate a schedule that deconflicts the moving of data.

It is a well-known fact that the DoD relies heavily on computer systems to conduct military and day-to-day support operations. The DoD is striving to materialize the vision of Network Centric Warfare/Network Centric Operations (NCW/NCO) by networking all available sensors in the battlefield to the Soldiers, Sailors, Marines and Airmen conducting the mission. The NCW/NCO concept allows the DoD to translate information into combat power through timely dissemination of this information. This information is used in the battlefield by warfighters to achieve shared awareness, increased survivability, higher operation tempo, greater lethality, improve speed of command and certain degree of self-synchronization [35].

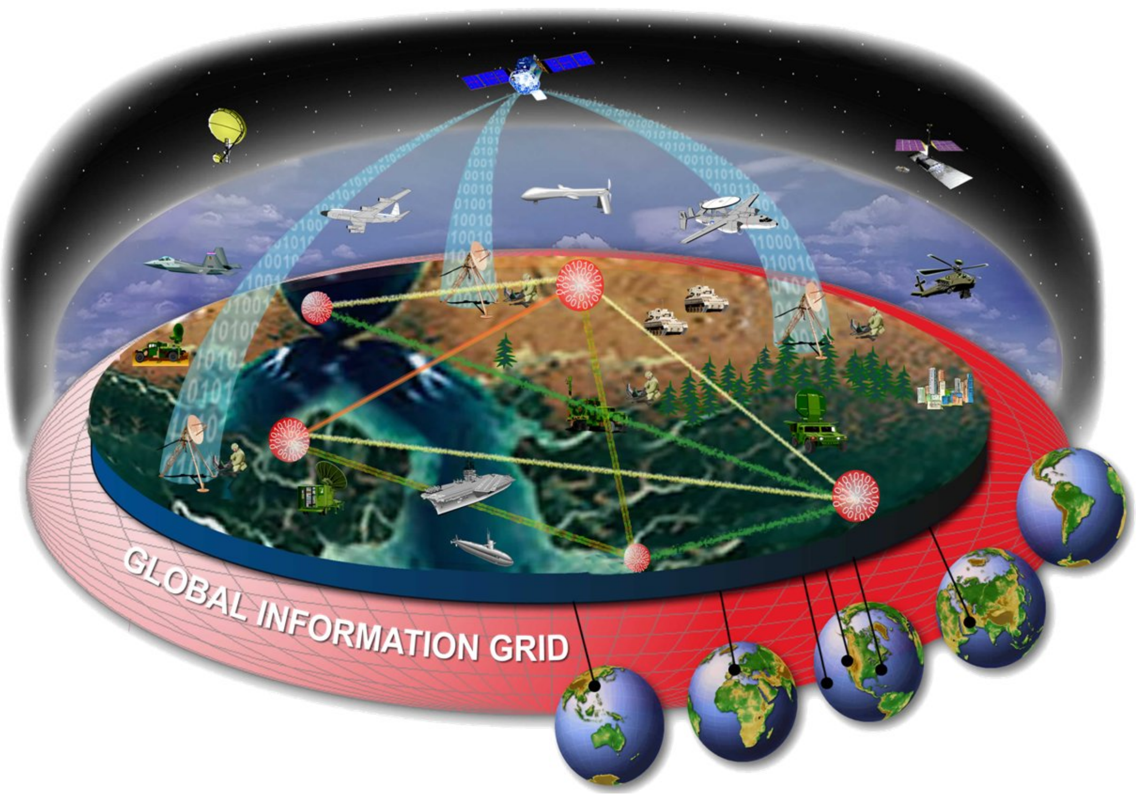

Figure 1.1: Global Information Grid [25]

In order to maintain its advanced warfighting capabilities, the DoD is always exploring new alternatives to improve its networks in ways that foster information superiority, decision superiority, and full-spectrum dominance. As part of this objective the DoD is currently implementing the Global Information Grid (GIG): a globally interconneted network that provides capabilities for collecting, processing, storing, disseminating, and managing information on demand to warfighters, policy makers, and support personnel. GIG capabilities will be available from all operating locations: bases, posts, camps, stations, facilities, mobile platforms, and deployed sites [25]. The GIG concept is depicted in Figure 1.1. One of the primary objecties of the GIG is to to ensure that the networks used to support military operations perform optimally so that they can provide feedback about the mission execution in a timely manner.

1.1 Objectives

The goal of this research is to develop an algorithm that will maximize bandwidth utilization in a military airborne network environment. This algorithm will use an offline environment to compute optimum routes prior to mission execution. The algorithm seeks to take advantage of the data collected during the mission planning phase for military tactical environments such as the ATO generation process. The algorithm will determine network packets priority and scheduling for all participating nodes in order to maximize utilization. A prioritized list of unicast and multicast network packets necessary to conduct the mission will be stored in an Information Exchange Requirements (IER) database.

The IERs entries contain periodic messages such as: Blue Force Tracking, Battle Damage Assessment, Command and Control messages, etc. The algorithm will use the IERs available during the mission planning stage as input in order to compute an optimum schedule with respect to available and accessible bandwidth.

1.2 Contributions

This thesis combines several prior research efforts in a novel way to achieve results not previously realized. Some of the components are well studied and documented, but they have not been integrated in a mobile environment.

Kalman filters have traditionally been used in many applications to estimate the state of a linear system (e.g., weapon guidance system). Recently, researches have applied Kalman filters to obtain network predictions a few seconds into the future within static networks. Network controls take advantages of these predictions to maximize throughput, but do not account for node mobility [12].

This research uses Kalman filters to obtain network predictions in a mobile network environment. Network controls are established to take advantage of these predictions in order to minimize network congestion while improving network throughput. Additionally, this research uses long-term predictions extracted from military mission planning documents to forecast network topology and traffic. These long-term predictions allow the network to be optimized prior to mission execution.

1.3 Thesis Overview

This chapter provided a brief introduction to the subject area of tactical airborne networks and highlighted some of the inherent challenges in this type of network environment. Additionally, this chapter outlines my research goals. An overview of the rest of the document is as follows:

- Chapter 2 provides an in-depth material review that show what research has already been conducted in this area. This literature review provides an introduction to pertinent concepts and terminology applied and referenced throughout this thesis.
- Chapter 3 describes the experimental methodology and details the approach used in conducting the experiments.
- Chapter 4 presents the results of the experiments conducted in a logical manner.
- Chapter 5 provides conclusions and recommendations for future work.

2 Literature Review

2.1 Chapter Overview

The objectives of this chapter are to provide the necessary background information to precisely define the problem and review the current state-of-the-art technologies contributing to my proposed solution. This chapter presents this background information using a top to bottom approach, beginning with Mobile Ad Hoc Network (MANET), network predictions, Network Tasking Order (NTO), Kalman filters, Network Weatherman, multi-commodity flows, High Assurance Internet Protocol Encryption devices and finally narrowing down to the specific focus of this research and how to use network predictions to improve network performance.

2.2 Mobile Ad Hoc Networks

A MANET is a collection of mobile nodes that generate a network automatically by creating several short-lived radio frequency links based on node proximity and mutual agreement [18]. MANETs rely heavily on wireless technologies to establish network connections. These connections are very unpredictable because links can be lost due to to blocking, obstacles or other types of interference.

Two nodes can establish communication directly in a MANET if the recipient is within the transmission range of the source node. However, if the source node cannot reach the recipient directly, the communication session is established via a series of intermediate nodes. For this reason MANETs are also called multi-hop packet radio. In this type of network nodes, in addition to acting as a host they also route network traffic for other nodes that are not within direct wireless range of each other. Every node in the network participates in a protocol that facilitates the discovery of multi-hop paths throughout the network to any other node [16].

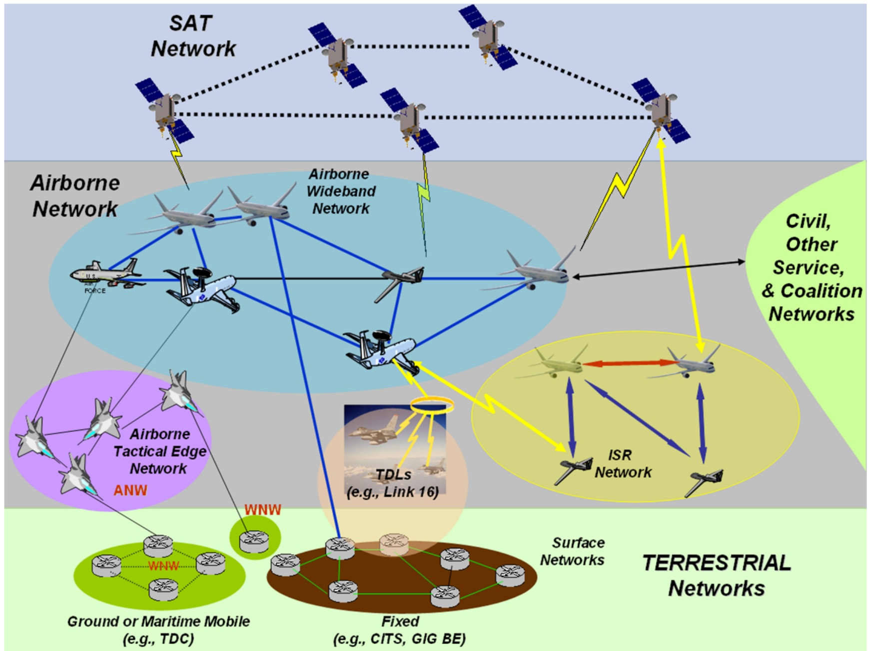

Figure 2.1: A Mobile Ad Hoc Network

The U.S. military forces utilize MANET for several different applications ranging from few static nodes to large scale, mobile, highly dynamic networks. Because of the broad range of applications It is not possible to design a MANET protocol that provides optimum performance in all environments. However, It is important to emphasize that any MANET protocol designed to provide optimum performance must take into consideration the following limitations inherent [31]:

- MANET topologies are highly dynamic and very hard to predict because the network connections are created based on the proximity of nodes that are constantly moving.
- The use of wireless transmission impose a severe limitation to the maximum link capacity that can be achieved when compared to wired networks.

- Physical security is hard to establish due to the wireless transmissions. Physical security concerns are mitigated by using encryption mechanisms to prevent eavesdropping and highly directional antennas are used to prevent jamming attacks.
- Due to the highly dynamic environment MANETs are normally affected with a high loss rate and high delays when compared to traditional networks.
- MANET nodes have a power source that is severely constrained such as batteries, generators, etc. This limitation drives important design considerations such as: CPU, memory, radio transmission, etc [8].

Conventional MANETs rely on routing protocols that require periodic advertisements that propagate throughout the entire network. These messages are broadcasted in order to distribute a copy of the latest routing information to all participating nodes/routers. MANET protocols that require routing information to be disseminated to every host in the network present several disadvantages. First, the distribution of routing information consumes bandwidth which sometimes is very limited in this type of network setting. Secondly, the repeated updates keep awakening nodes that may be operating using a constrained power source, e.g., battery, generators. Finally, this type of network protocols often cannot operate efficiently in an environment where the network topology is highly dynamic because they cannot respond quickly enough [17].

2.3 Ad Hoc On-Demand Distance Vector Protocol

The Ad Hoc On-Demand Distance Vector Protocol (AODV) is a MANET protocol with quick convergence that creates routes on demand that are loop free. AODV requires very low processing and memory overhead when compared to other MANET routing protocols while delivering excellent performance. AODV scales very well because it only needs to addresses to create a route: destination and next hop. AODV is very popular because of all of these features and is currently the most widely deployed MANET protocol [1].

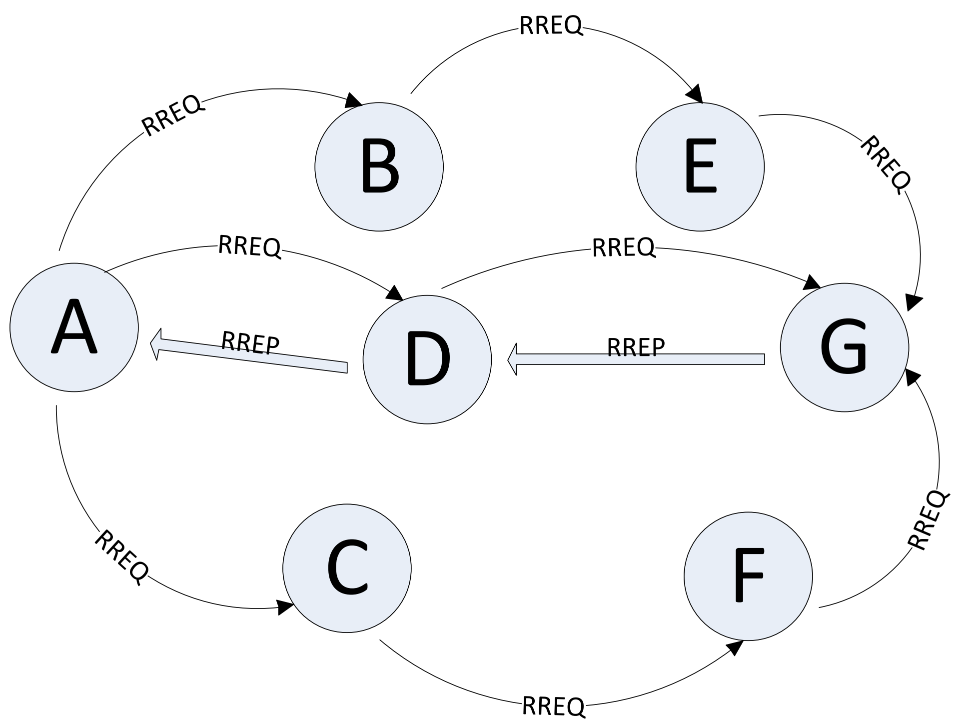

Figure 2.2: Ad Hoc On-Demand Distance Vector Protocol – Node *A* requests routing information for node *G*.

The AODV algorithm does not require bootstrapping and is capable of generate multi-hop routing between participating nodes in the network. Routes are created whenever a source node attempts to establish a communication with a node that is not in its routing table by broadcasting a Route Request Message (RREQ). The RREQ message is rebroadcasted to all participating nodes in the network until the message reaches the destination node. Intermediate nodes note the sender of each unique RREQ and ignore any duplicate RREQs to avoid generating loops. Once the RREQ message reaches the destination node, this node reply an unicast Route Response Message (RRER) to the source node with the routing information necessary in order to establish a backward path.

Each node keeps a record of successful RREQ/RRER pair in their routing table for future reference [4]. This process is illustrated in Figure 2.2.

There are some disadvantages associated with the use of AODV in some specific circumstances. First, AODV requires periodic beaconing to refresh network routes leading to unnecessary bandwidth consumption. Additionally, this routing protocol does not provide optimum performance in MANET environments where the network topology is highly dynamic because the network routes become stale quickly requiring several RREQ broadcast messages in order to refresh the routing paths [17].

2.4 Multi-commodity Flow

Traditional network environments use an implementation of a shortest path algorithm to generate optimum routes. However, this choice of algorithm does not achieve optimal results in a network environment where the nodes are constantly moving such as MANET. These limitations to shortest path algorithm usage are well known within academia:

> In a dynamic network environment under heavy traffic load, shortest-path routing algorithms, particularly those that attempt to adapt to traffic changes, frequently exhibit oscillatory behaviors and cause performance degradation [33].

A multi-commodity flow is a network flow paradigm with multiple sources and destinations that maximizes the amount of flow traveling from the various sources to their corresponding destinations subject to the capacity constraints. A multi-commodity flow instance can be defined as a set of ordered pairs of vertices (S_1, T_1), $(S_2, T_2) \ldots (S_K, T_K)$, where each pair (S_i, K_i) represents a commodity with source S_i and target T_i. For each commodity (S_i, K_i) a non-negative demand d_i is specified [27]. The objective is to maximize the amount of flow traveling from the sources to the corresponding destinations as shown in Figure 2.3.

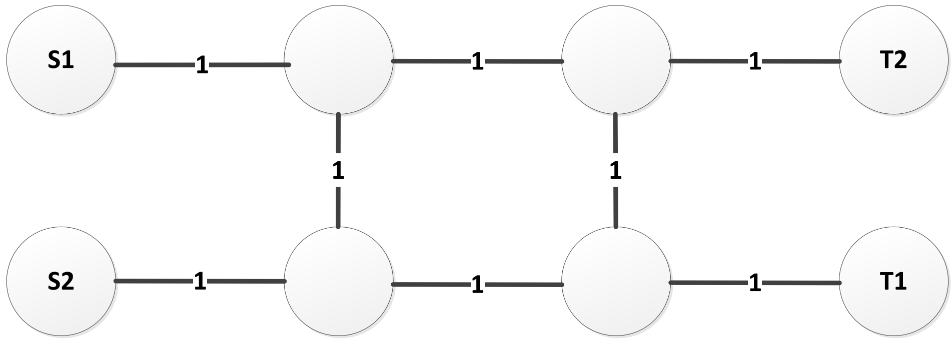

Figure 2.3: Maximum Multi-commodity Flow Problem $d_1 = d_2 = 1$ [27]

There are several algorithms that can solve some specific instances of the multi-commodity flow problem in polynomial time using linear programming if we allow fractions for the flow demands [27]. Figure 2.4 presents a solution to the multi-commodity flow previously described in Figure 2.3 using fractional flows.

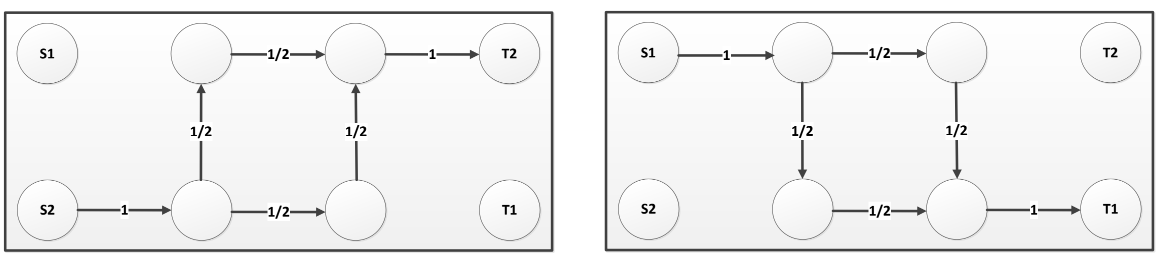

Figure 2.4: Solution to a 2-commodity flow problem [27]

Normally a network routing solution based on multi-commodity flow is problematic because this paradigm is intractable if we constraint the flow demands to be integers. Shimon Even from The Weizman Institute of Science demonstrated in 1975 that the multi-commodity integral flow problem is Nondeterministic Polynomial-time (NP) complete even if the number of commodities is two [6]. The computing resources necessary to solve the problem in polynomial time using linear programming and allowing fractional flows make these implementations impractical. In reality, an algorithm that computes the fractional solution to within one percent is normally adequate for a practical implementation that can be used to determine network routes [10]. It is possible to obtain an approximation of the optimum solution arbitrarily close in polynomial time much faster than algorithms based on linear programming [11].

2.5 High Assurance Internet Protocol Encryptor

Military networks are used to transmit secure and non-secure information. Secure information utilizes a dedicated network infrastructure that is physically separated from any other network that does not have the same network classification. Secured networks are referred to as red networks. Networks used to transmit non-secure information are referred as black networks. It is assumed that any traffic transmitted over black networks can be monitored. The security protocols used to implement red networks are designed to prevent monitoring traffic by unauthorized parties.

It is possible to transmit red network traffic through an unsecured channel (such as a black network) when High Assurance Internet Protocol Encryptor (HAIPE) products are used to encrypt the information. HAIPE devices essentially establish a tunnel to transmit red network traffic over a black network. The Committee on National Security System dictates that HAIPE should establish the network tunnel as follows:

> Use the Internet Engineering Task Force (IETF) Encapsulating Security Payload version 3 (ESPv3) to encapsulate plaintext IPv4 or IPv6 traffic. HAIPE uses the IETF's Simple Network Management Protocol version 3 (SNMPv3) to support over-the-network management and the IETF's Routing Information Protocol version 2 (RIPv2) and Routing Information Protocol Next Generation (RIPng) to provide the HAIPE local discovery capability. The IETF has defined a cryptographic transform based upon Advanced Encryption Standard (AES) and Galois Counter Mode (GCM) [2].

Such usage of IP encryption to secure network traffic introduces several challenges to a MANET protocol that makes routing decisions based on NTO's inputs. HAIPEs add significant overhead to the network traffic that passes through black network. Specifically, it consumes a significant portion of the bandwidth for peer discovery and establishing security associations with packet headers added to support these various security services. HAIPEs also affect the end-to-end performance by introducing processing delay for encryption and decryption of packets [21].

Another challenge introduced by HAIPE devices pertains to the information allowed to be exposed outside of the network tunnel that it established. Namely, the only information that can be exposed to the unprotected side of the network is source and destination HAIPE device information. This is done primarily to prevent information leakage such as identification of nodes, network topologies, etc that exist behind HAIPE devices (please refer to Figure 2.5). As a result of this strict requirement, the priority of traffic within HAIPE devices must be uniform.

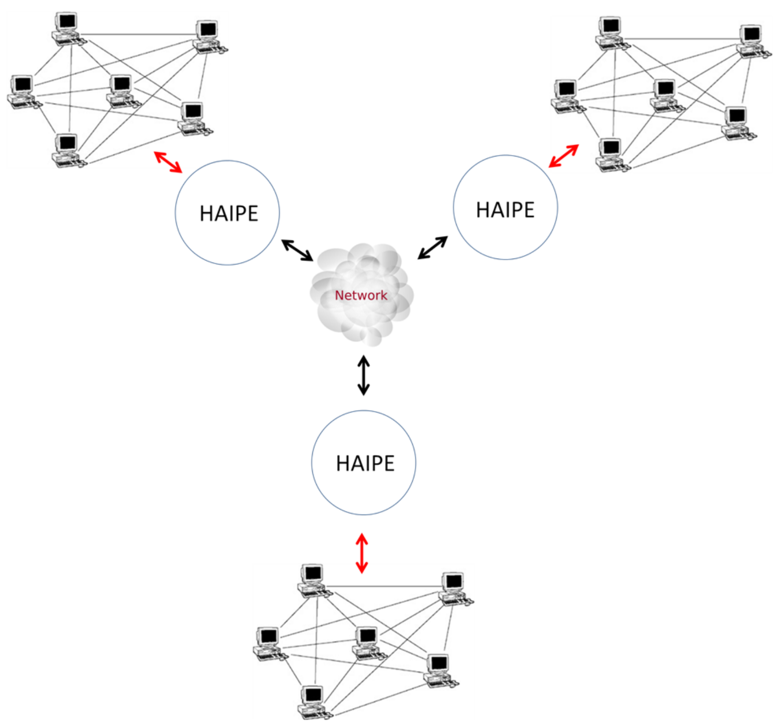

Figure 2.5: Red Network Traffic Protected With HAIPE Devices

In order to implement a MANET protocol that makes routing decisions based on predictions, it is necessary to share protocol information as well as some key performance metrics among routers, such as queue sizes. It is possible to implement such a protocol in a High Assurance IP network if we constrain all routing information to the protected side of the network tunnel. This task can be accomplished with out-of-band channel using TCP/IP within the red network to exchange protocol information. This technique will allow optimization and prioritization within the collective red network. As a result, only standard optimization techniques can be accomplished within the black routes that are going to be used to transmit the encrypted data.

Networks predictions can be used to optimize complex network environments that include red network traffic encrypted with HAIPE devices as well as normal black network traffic as shown in Figure 2.6. Nodes transmitting unsecured traffic can exchange the required protocol information and performance metrics necessary to implement optimized network protocol using out-of-band channel encrypted with commercial encryption standards such as Internet Protocol Security (IPSec). Using this technique, red-network traffic is going to be optimized and prioritized independently from nodes transmitting black traffic.

2.6 Network Predictions

Traffic prediction plays an important role in network control and protocols. It is possible to implement automatic network controls that adjust network routes in order to minimize congestion based on accurate projection of the future state of the network. In order to implement such a system the prediction interval should be large enough to offset any delays caused by traffic measurements, network topology changes, modify traffic priority, etc.

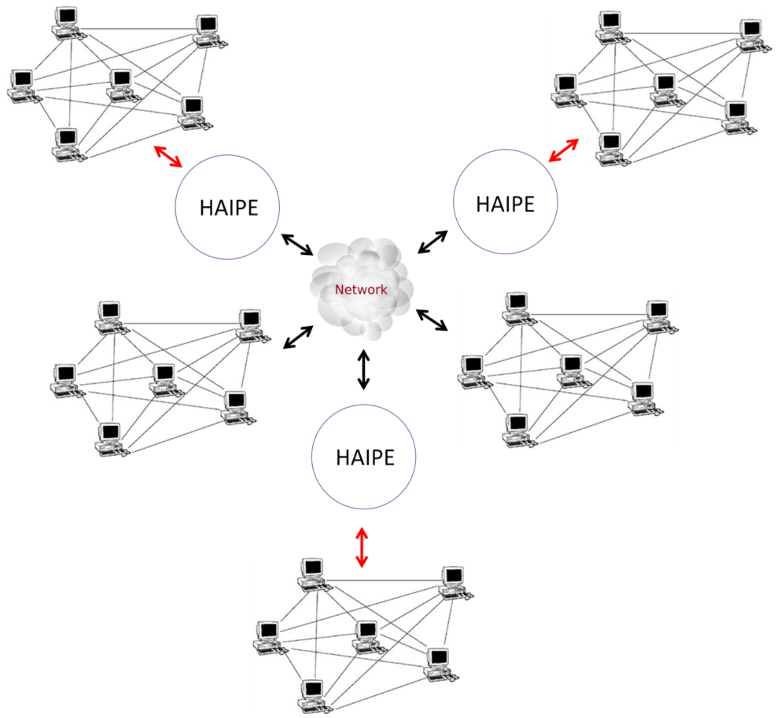

Figure 2.6: Red Traffic/Black Traffic Competing For Resources

A significant part of the bandwidth in MANETs environments is consumed by the messages broadcasted to advertise topology changes, network routes, traffic changes, etc. A solution to this problem is to predict the future state of the system far enough in advance so that network controls can avoid congestion and be able to utilize the available network capacity efficiently.

It is important to determine how far into the future can network traffic be predicted with confidence because the prediction accuracy decreases significantly as the prediction interval is increased. Automatic network controls making decisions based on erroneous predictions can deteriorate the performance of the overall network [26].

2.6.1 Kalman Filter. In 1960 Dr. Rudolph. E. Kalman introduced his famous concept of how to compute an optimum estimate for linear filtering and prediction. Since that time, this algorithm has been subjected to extensive research and has proven to be extremely practical and useful [5]. This process can be used to *filter* noise from discrete data by computing a least-squares curve-fit [34]. Since the filter minimizes the error with every data sample supplied as exemplified in Figure 2.8, its estimate converges to the actual state of the system. It is also important to note that the prediction of the filter in a linear system is significantly better than any individual measurement after just a few iterations as shown in Figure 2.7.

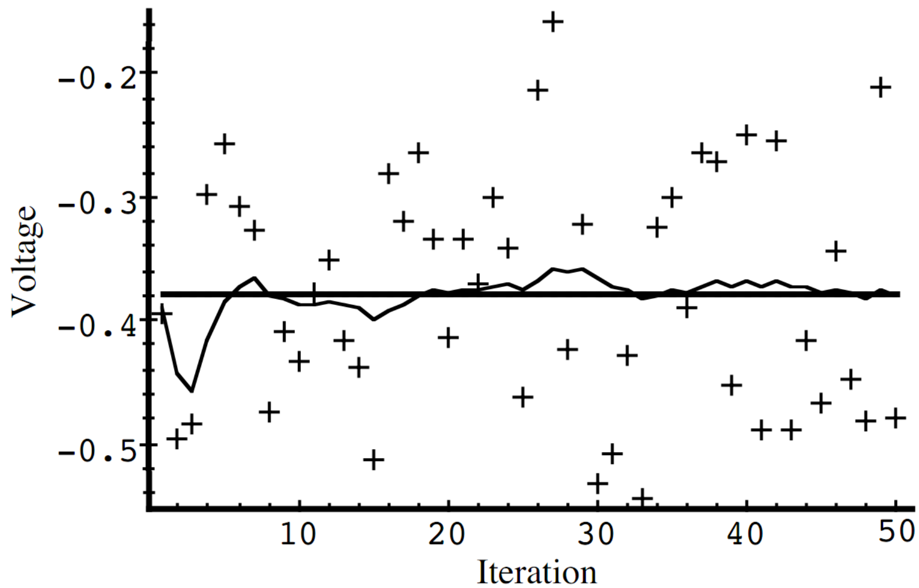

Figure 2.7: Kalman Filter Performance. The true value of the random constant $x = -0.37727$ is given by the solid line, the noisy measurements by the cross marks, and the filter estimate by the remaining curve [34].

Kalman filters are used in several applications to determine how a system can deal on-line with uncertainty in an optimal way. For example, they have been used to integrate several sensors such as Inertial Navigation Systems (INS), Global Positioning System (GPS) and optical seekers to predict the position and velocity of a weapon system achieving remarkable results. Because of the low computational cost and low memory

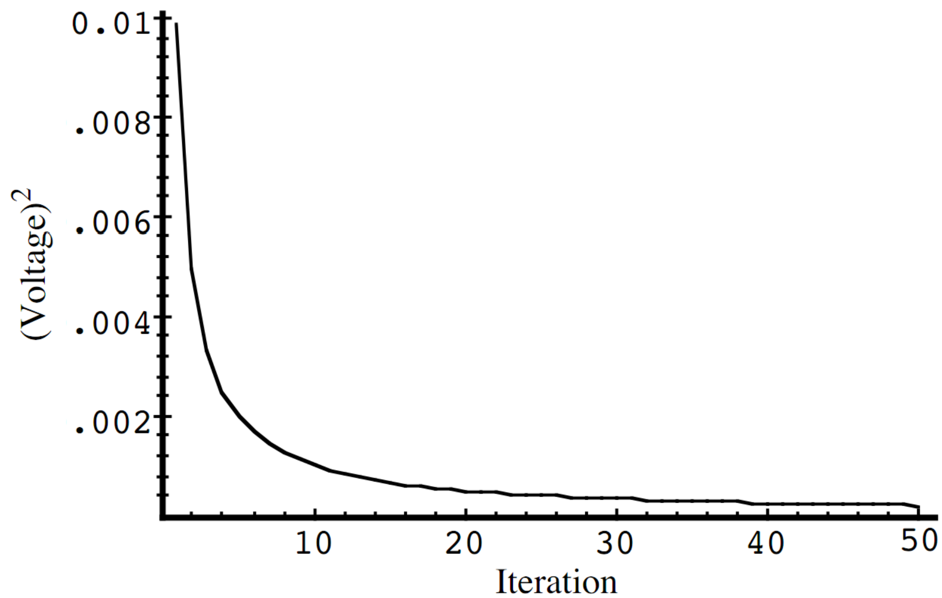

Figure 2.8: Kalman Filter Error Estimation. After 50 iterations, the initial error covariance P_k choice of 1 has settled to about 0.0002 ($Volts^2$) [34].

requirements it has been possible to use Kalman filters in consumer-grade portable GPS car navigation systems as well.

Kalman filtering is a fairly direct and simple process. The filter is a recursive algorithm that analyzes data in a linear system in which the system's state is observed through a set of discrete measurements. The filter works using feedback and control mechanism; it computes the estimate of the system and then obtains feedback in the form of a *noisy* measurement. Therefore, the filter uses two set of equations: time-update related and measurement-update related. Time-update equations are used to project the future state of the system while the set of measurement-update equations are used for the feedback [34] as illustrated in Figure 2.9. One of the reasons why Kalman filtering is so popular is because the Time-update equations can be used to compute an optimum estimate of past, present and future states of the system by aggregating the information obtained from the discrete measurements.

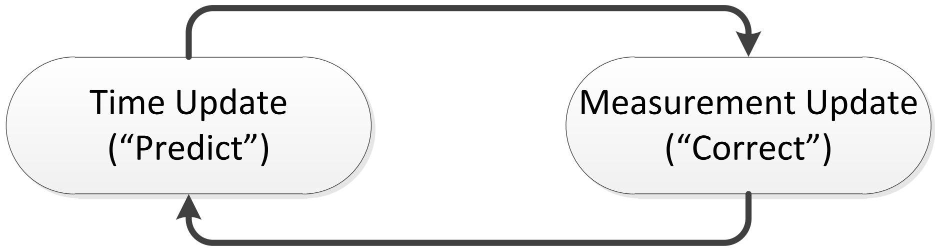

Figure 2.9: The ongoing discrete Kalman filter cycle. The *time update* projects the current state estimate ahead in time. The *measurement update* adjusts the projected estimate by an actual measurement at that time [34].

Kalman filtering can be used to predict the future state of a network based on past performance using minimum computational resources. Estimating the future state of a network by batch processing all past measurements is a problem with increasing computational load; and given sufficient time the problem may become intractable. Therefore, Kalman filters are extremelly atractive to solve this problem because it is an online recursive algorithm that takes a single past measurements at a specified interval to compute future estimates of the network.

Kalman filters are state estimators for linear systems, and the performance of the network system cannot be described by a linear set of equations. Therefore the quality of the estimates for the future state of the network that we can obtain from a Kalman filter strongly depends on how well linearized are the measurement functions and how they take linearization errors into account [20].

2.6.2 Network Weatherman. Nathan Stuckey developed the idea of *Network Weatherman*, an algorithm that can be used to predict future status of the network based on the network's present condition. Network Weatherman is implemented using a Kalman filter to predict future queue sizes in a network. This algorithm can accurately predict queue size several seconds into future as illustrated in Figure 2.10. These queue sizes predictions can be used in network control algorithms to optimally manage the network, thus optimizing metrics of interest such as delay or throughput [30].

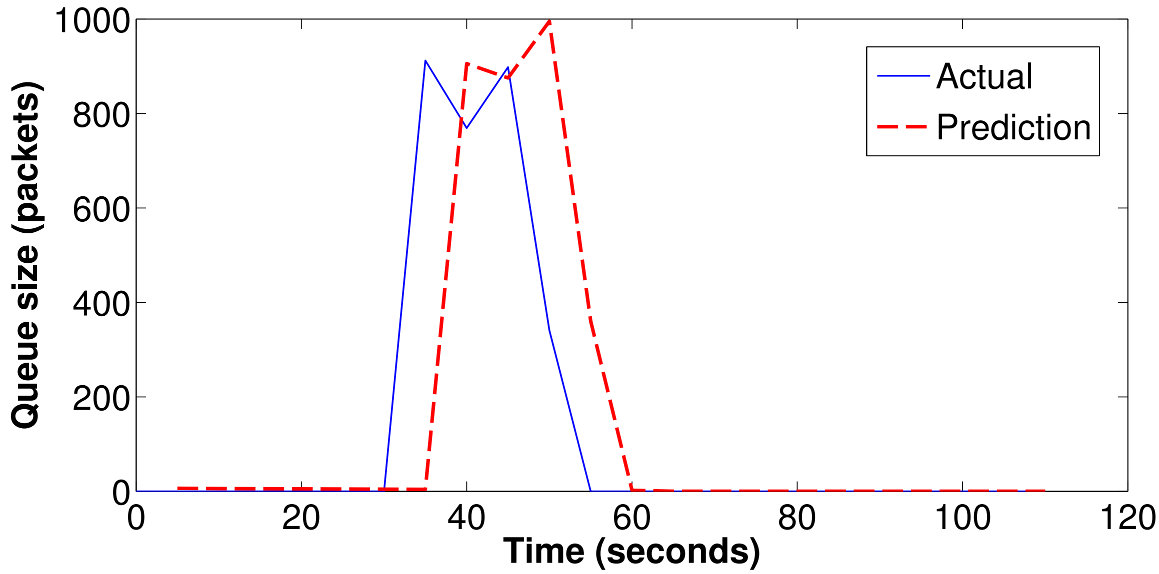

Figure 2.10: Network Weatherman With 5 Seconds Predictions

James Haught performed several simulations recently using a static scenario to validate the effectiveness of Kalman filters predicting network queue sizes. These simulations consisted of fourteen nodes with five Kalman filters placed at key locations in the network as shown in Figure 2.11. Each node used a drop-tail queue and had a maximum queue size of one thousand packets. Each node was assigned various TCP and UDP flows which had an exponential arrival rate. The Kalman filters were able to accurately predict the network flow several seconds into the future, making a prediction every second for the five hundred seconds of simulation time. However, it was noted that predictions too far into the future were inaccurate [12].

Haught performed additional experiments using the static scenarios described in Figure 2.11 adding a Dynamic Routing Queue Controller (DRQC) that managed the network traffic into the network. The DRQC adjusted network traffic based on current

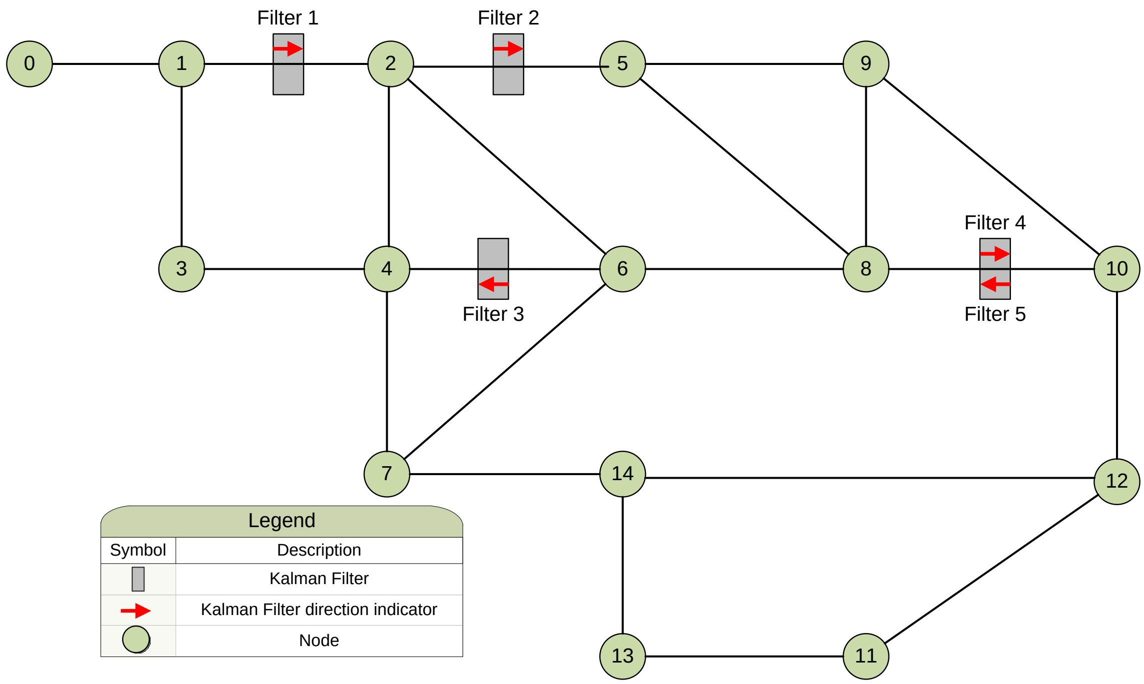

Figure 2.11: Simulation Diagram for Kalman Filter Validation

queue sizes as well as predicted queue sizes and network flows. A significant improvement was noted in the overall network throughput when the DRQC was introduced into the network measured by a drastic change in network congestion. It was also noted that the accuracy of the queue size prediction can affect significantly the performance of the DRQC [12].

2.6.3 Network Tasking Order. The Air Tasking Order (ATO) is a human/machine readable document created by Joint Air Planners that defines the daily schedule of air related operations. ATOs specify targets, callsigns, air controlling agencies, type and number of aircraft allocated to a target, mission types, routes (including time and position), frequencies in use, etc [23]. The information contained in the ATO is enough to determine what type of network support is going to be necessary to accomplish the mission. A document analog to the ATO named the Network Tasking Order (NTO) could be created that specifies network assets and operations extracted from the ATO.

One way to illustrate the benefits of creating an NTO is with the following example. In this example we have two nodes: Source 1 and Source 2. Each node has data to transmit to the same destination node. Source 1 is transmitting a video at 30.6 Kilobits per second (Kbps) and based on the requirements specified in the ATO it has been determined to be a high priority transmission. Source 2 is collecting data from a remote sensor at 7.2 Kbps and based on the ATO requirements it has been classified as low priority. Both sources are connected to the destination node via the same router. There is a single link between the router and the destination node. The capacity on this link can only accommodate a maximum data transfer of 36 Kbps. From the information above, the combined data rate for both sources is 37.8 Kbps. Hence, the available bandwidth is not enough to allow data transmitted from both sources simultaneously to be delivered to the destination node as shown in Figure 2.12.

One of the challenges typically encountered in MANET using traditional routing protocols is highlighted above where not all of the available resources are utilized. However, using information collected from the NTO, an optimal alternative topology can be generated. The NTO provides enough information to leverage networking resources not normally considered by traditional routing protocols. For example, in the previous scenario there is an airborne platform that can be used to ferry data between Source 2 and the destination.

The airborne platform is on a thirty minute orbit track that places it within range of the Source 2 node forty percent of the time, within range of the Destination node also forty percent of the time, and out of range of both nodes the remaining twenty percent of the time. Based on this orbit track, Source 2 is never aware that a connection exists between the airborne platform and the destination node. As far as Source 2 is aware, no

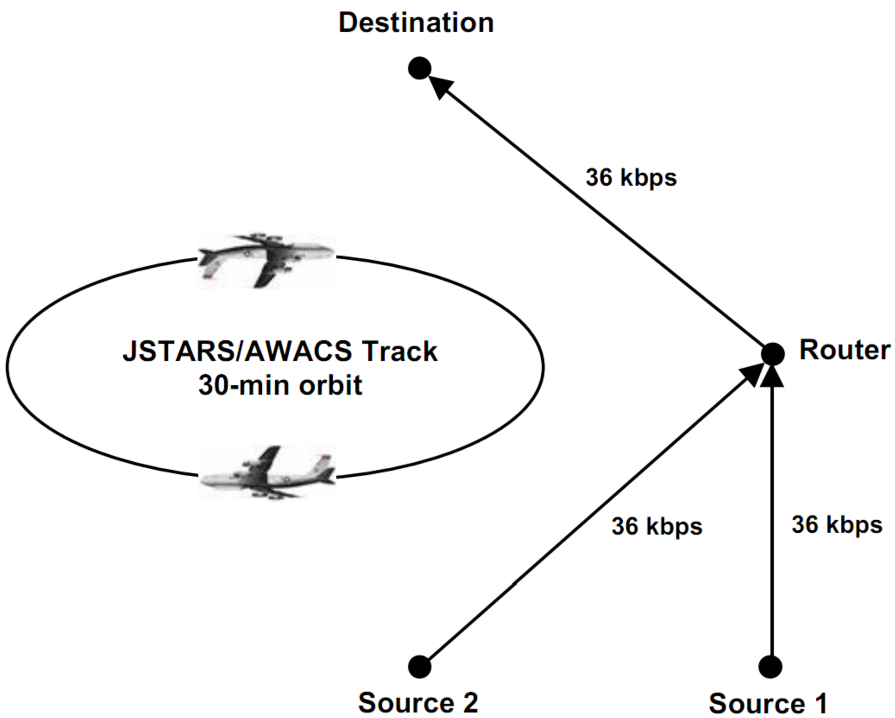

Figure 2.12: Scenario without an NTO [3]

connection ever exists between the airborne platform and the destination, and it will not be considered as a possible path to deliver data to the destination node.

Leveraging information from the NTO, data from Source 2 to the destination is forced through the airborne platform. This allows Source 1 to use its existing connection to the destination via the router as dedicated link as shown in Figure 2.13. This solution ensures that Source 1 data is delivered as fast as possible which is consistent with its high priority classification.

2.7 Summary

This chapter presented the recent research and fundamental concepts in the areas of MANET, Kalman Filters and HAIPE devices. Additionally an explanation of ATO and NTO interaction was introduced. It is now time to construct an environment in which

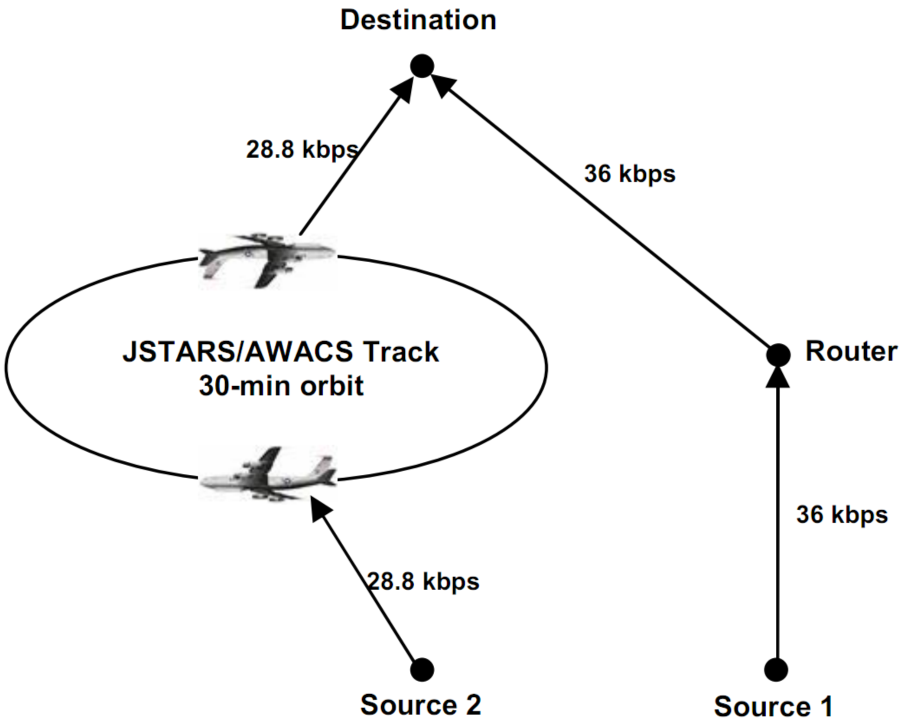

Figure 2.13: Scenario without an NTO [3]

exploit the information obtained via the NTO process to optimize MANETs for throughput while maintaining a quality of service.

3 Methodology

3.1 Introduction

This chapter outlines the methodology used to determine the applicability of the NTO concept to improve performance when compared with traditional MANET routing protocols. This chapter outlines the goals and hypotheses of this research, elaborates on the problem, applies knowledge of the environment and describes the measures of merit on which the results of the algorithm will be judged. An outline of the experiments to be performed is given. The expected results are given and the expected performance factors are stated.

3.2 Research Objectives

The objective of this research is to develop an algorithm that capitalizes on the information obtained via the NTO process. The algorithm will compute optimum routes based on predicted network traffic and topology. This part of the algorithm will be executed in an offline environment prior to mission execution. The routes generated in this process will be preloaded in all participating nodes.

The NTO process provides estimates of the network performance over a course timeframe based on limited information available several hours before mission execution. It is possible to encounter significant topology or network traffic deviation from these estimates during actual mission execution. Therefore, the algorithm will have agents monitoring network traffic online in order to detect deviations from the NTO process and generate network routes that minimize the impact of inaccurate network predictions.

3.3 Research Hypothesis

There are four hypotheses that will be considered throughout this research:

- The NTO concept is applied to obtain long-term predictions for network traffic and topology that can be used compute optimum routes.
- The number of packets dropped will be lower when the NTO concept is applied in conjuction with the online agent when compared to AODV routing protocol.
- The overall network throughput will be higher when the NTO concept is applied in conjuction with the online agent when compared to AODV routing protocol.
- The throughput of high priority traffic will be higher when the NTO concept is applied in conjuction with the online agent when compared to AODV routing protocol.

3.4 Measures of Merit

The measures of merit are: the algorithm ability to minimize packet loss, improve network throughput without affecting the overall end-to-end delay. Packet loss will be measured by the amount of dropped packets throughout mission execution. Network throughput will be measured by the average KBps during mission execution. End-to-end delay will be measured by the amount average time in seconds that it takes for packets to travel from source to destination.

3.5 Caspian Sea Scenario

The Caspian sea scenario was developed by Major Michael Davis and Major Reginal Kabban to develop metrics that can be used to measure the performance of an information collection and exploitation system [19]. The scenario was later enhanced to include several airborne platforms. This enhancement relies heavily on wireless communications, making it suitable to therefore conduct MANET simulations.

The scenario describes a notional conflict that occurs between two countries, Azery and AnFar vying for a region. AnFar military forces occupy the disputed area, forcing

Azery forces to defend the region. The US military responds to diffuse the situation and prevent conflict escalation.

The U.S. response modeled in the simulation requires the deployment of intelligence collection equipment such as UAVs, unattended ground sensors, etc. The simulation includes several aerial platforms such as JSTARS, F-22s, UAVs, etc. The scenario uses a MANET to disseminate information collected by sensors and airborne platforms to decision makers.

The Caspian Sea Scenario is appropriate for this research because it provides a detailed description of the mobility for the different airborne platforms. The scenario also provides guidelines for network traffic generation. Generating all simulations based on this scenario facilitates future research as well because it provides a baseline that can be used to compare the performance of various algorithms.

3.6 Network Tasking Order Model Implementation

To realize an improvement on network performance based on NTO predictions, it is necessary to discretize individual node's mobility information. This technique permits the prediction of network topology throughout mission execution as identified in the NTO. This research implements this process using the Caspian Sea scenario in two steps. The first step consists in computing the individual nodes position at discrete time intervals. The second step computes the distance among all participating nodes at discrete time intervals.

3.6.1 Compute Nodes' Position. The NTO generates predictions for the location of the different participating airborne platforms by parsing the ATO. Figure 3.1 illustrate the input provided to the NTO process. In this sample ATO, the mission is divided in six lines. The first line of the ATO identifies the country in charge of the mission, which in this instance is the U.S. The second line identifies the military service in charge of the operation, which in this case refers to the Air Force. The third line identifies the 555

Fighter Squadron located in Spangdahlem Air Base (ETAD) as the unit in charge of the mission. The fourth line specifies the mission number as C2342 and the mission type as Combat Search and Rescue. The following lines specify the coordinates for the location of the Aircraft at specific times. For example, the line: `[/01/-/294248N0473106E/241200ZJAN]` specifies the aircraft location as 29° 42' 48" North 47° 31' 06" East, at 1200 Zulu time on 24 January [9].

```
1-TSKCNTRY/US//
2-SRCVTASK/F//
3-TASKUNIT/555FS/ICAO:ETAD//
4-AMSNDAT/C2342/CSAR//
    /DE/TGT-ID/LOCATION/TOT
    /01/-/294248N0473106E/241200ZJAN
    /02/-/294300N0473896E/241215ZJAN
    /03/-/294300N0473805E/241233ZJAN
    /04/-/294236N0473106E/241303ZJAN
5-MSNACFT/1/ACTYP:F16C/SANDY01/2MK-82
    /1654/3322//
6-AMSNLOC/AGL200/1//
```

Figure 3.1: Air Tasking Order Example [9]

The NTO utilizes this information to generate a forecast of the trajectories for all participating nodes. The location of all nodes at a specific time during mission execution can be predicted using this model. Table 3.1 illustrate this concept by showing the location of eighteen nodes participating in a MANET 8100 seconds after mission start.

It is advantageous to generate several tables at specified time intervals to model the dynamics of the network prior mission execution. The Caspian Sea scenario specifies the mobility of the nodes using this approach. This specific scenario contains 150 tables, one table per minute for the 150 minutes of simulation time.

Table 3.1: Node's Position – The information presented in this table is notional and it does not correlate to the Caspian Sea scenario or real events.

Node_ID	Time (Seconds)	X Position(Feet)	Y Position(Feet)	Z Position(Feet)
1	8,100	174,449	709,160	6,096
2	8,100	436,748	769,637	6,705
3	8,100	577,265	777,840	6,096
4	8,100	788,872	609,752	6,096
5	8,100	1,081,194	695,136	6,096
6	8,100	333,360	348,673	18,288
7	8,100	763,127	623,568	18,897
8	8,100	494,884	467,064	3,048
9	8,100	851,892	550,564	3,657
10	8,100	872,555	1,031,564	4,572
11	8,100	923,453	1,021,378	5,486
12	8,100	972,761	1,008,174	4,572
13	8,100	872,555	168,532	4,572
14	8,100	923,453	178,718	5,486
15	8,100	972,761	191,922	4,572
16	8,100	296,320	694,500	0
17	8,100	481,520	666,720	0
18	8,100	1,037,120	972,300	0

3.6.2 Determine Distance Among Nodes. The distance between two nodes is computed utilizing the Euclidian formula as follows: Let node A's position = (A_x, A_y, A_z) and node B's position = (B_x, B_y, B_z) the distance between nodes A and B is $\sqrt{(A_x - B_x)^2 + (A_y - B_y)^2 + (A_z - B_z)^2}$. The distance among all nodes is computed using two nested loops that evaluate each node with all other nodes in the simulation. This process generates a Distance table as illustrated in Table 3.2.

The angle between two given nodes can be computed as follows: $\theta = \arccos \frac{(A \cdot B)}{|A||B|}$. The distance and angle between two nodes can be used to forecast connectivity and link capacity. This research assumes that all nodes are equipped with isotropic antennas; therefore, the angle between nodes is not used to compute node connectivity or link capacity.

Table 3.2: Distance in Miles Among Nodes – The information presented in this table is notional and it does not correlate to the Caspian Sea scenario or real events.

Nodes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	44	67	102	149	65	98	66	115	127	134	140	145	151	157	20	51	148
2	44	0	23	64	107	71	59	51	77	84	90	97	122	126	130	26	19	104
3	67	23	0	44	84	81	40	53	59	64	70	75	111	114	116	48	24	82
4	102	64	44	0	50	86	5	54	14	71	71	72	74	74	75	82	51	72
5	149	107	84	50	0	136	54	104	45	65	60	55	93	89	85	129	99	46
6	65	71	81	86	136	0	84	33	92	143	147	151	94	101	108	57	58	155
7	98	59	40	5	54	84	0	51	19	70	71	72	77	78	79	78	47	73
8	66	51	53	54	104	33	51	0	60	112	115	119	79	85	91	50	33	122
9	115	77	59	14	45	92	19	60	0	79	78	78	63	62	62	94	64	76
10	127	84	64	71	65	143	70	112	79	0	9	17	142	141	139	110	88	29
11	134	90	70	71	60	147	71	115	78	9	0	8	141	139	137	116	93	20
12	140	97	75	72	55	151	72	119	78	17	8	0	139	137	134	123	98	12
13	145	122	111	74	93	94	77	79	63	142	141	139	0	9	17	128	104	135
14	151	126	114	74	89	101	78	85	62	141	139	137	9	0	8	134	108	132
15	157	130	116	75	85	108	79	91	62	139	137	134	17	8	0	139	112	129
16	20	26	48	82	129	57	78	50	94	110	116	123	128	134	139	0	31	130
17	51	19	24	51	99	58	47	33	64	88	93	98	104	108	112	31	0	104
18	148	104	82	72	46	155	73	122	76	29	20	12	135	132	129	130	104	0

3.7 Offline Algorithm To Implement Network Tasking Order Process

3.7.1 Introduction. The offline algorithm exploits the information available prior mission execution obtained using the NTO. The algorithm generates optimum routes for the topology and network traffic identified by the NTO. Network routes generated using this process are preloaded to all participating nodes in the MANET prior to mission execution. The main functionality of the algorithm is implemented in two key processes: Predict Network Topology and Compute Maximum Concurrent Multi-commodity Flow.

3.7.2 Predict Network Topology. This process takes as input tables with distances among all nodes as illustrated in Table 3.2. The output of this process is the forecasted network topology including connectivity among nodes and link capacity. Several tables are required to model a complete mission as nodes are constantly moving. The algorithm generates a new topology for each input table. The Caspian Sea scenario is modeled using 150 topologies, one per minute for the 150 minutes of simulation time.

Military MANETs normally use different types of radios to establish network links. The type of radio used depends on the airborne platforms executing the mission. The data rates that these radios can support are very diverse. Some radios support a maximum of 10 kilobits per second (kbps) while others can support up to 8 Megabytes per second (MBps) [29]. This research assumes that all participating platforms in the Caspian Sea scenario have a radio capable of transmitting and receiving up to 8 MBps. Implementation guidelines for the Caspian Sea scenario indicate that the researcher specifies the type of radio technology used as a backbone for the MANET.

Most Radio Frequency (RF) based networking devices are capable of adjusting the data rate in order to increase connectivity range. Radios with this capability implement several different modulation techniques. The radio selects a modulation scheme optimized for increased data throughput when operating in a high signal to noise ratio (SNR). However, a low data rate modulation is used in low SNR environments, because this type of modulation provides better performance for these conditions. Radios with automatic rate adaption select the modulation scheme that provides the optimum data throughput for a given channel condition [13]. The degradation of performance for a given modulation scheme is dramatic once the SNR drops below a specified range as illustrated in Figure 3.2.

Most military radios have a maximum range of 75 kilometers (approximately 45 miles) line of sight [29]. This research assumes that all platforms are using a radio with a maximum range of 45 miles. The algorithm generates links prediction based on the distance of the nodes. The specific implementation uses two nested loops to compare the distance of every node against every other node and create links prediction as follows:

- If the distance is less than 15 miles the algorithm predicts that there will be network link between those two nodes with a capacity of 8 MBps.

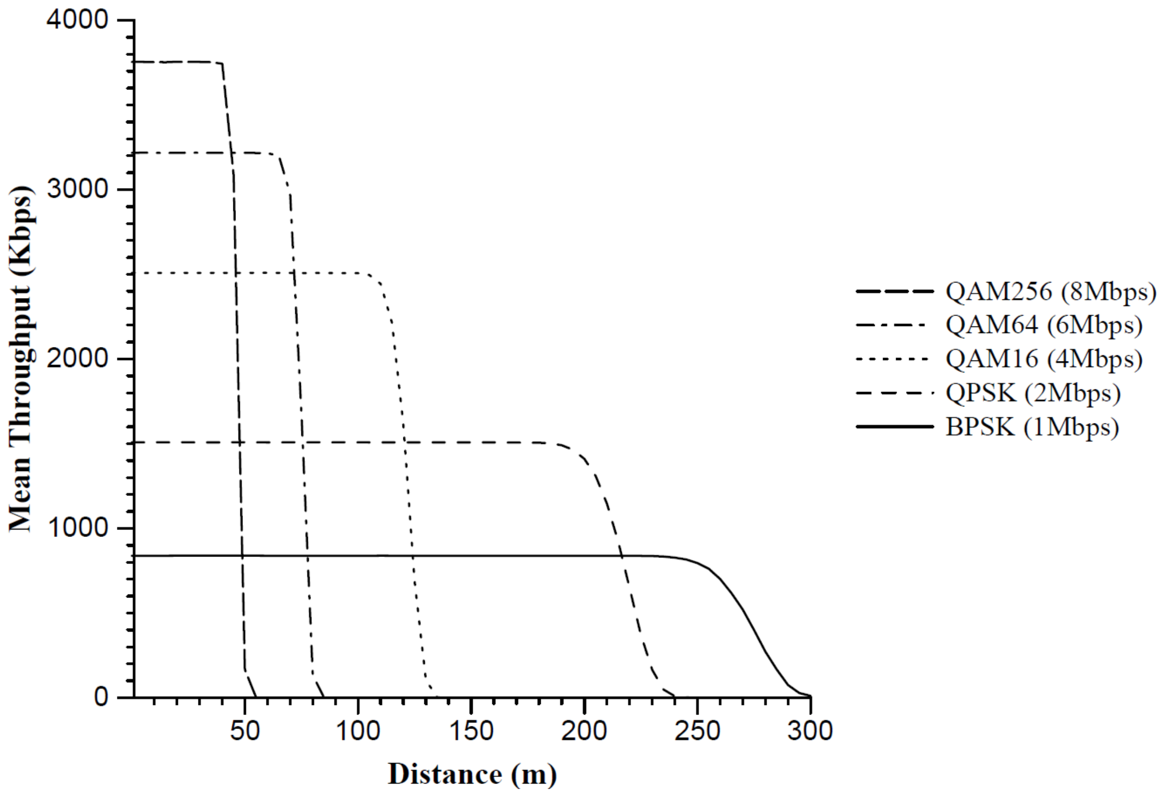

Figure 3.2: 802.11 Study of Throughput Versus Distance For Wireless Devices [13]

- If the distance is less than 25 miles the algorithm predicts that there will be network link between those two nodes with a capacity of 4 MBps.
- If the distance is less than 45 miles the algorithm predicts that there will be network link between those two nodes with a capacity of 2 MBps.
- If the distance between the nodes exceeds 45 miles, the algorithm will generate a prediction that the two nodes are out-of-range.

The mobility of the nodes specified in the Caspian Sea Scenario is such that sometimes clusters are formed because the nodes are over fifty miles apart. The Breadth First Search (BFS) algorithm illustrated in Figure 3.3 is used to identify clusters. The algorithm picks the first node in the scenario and performs a BFS. Any node identified as

Table 3.3: NTO Prediction For Network Topology in MBps– The information presented in this table is notional and it does not correlate to the Caspian Sea scenario or real events.

Nodes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1		2														4		
2	2		4													2	4	
3		4		2			2										4	
4			2				8		8									
5									2									
6								2										
7			2	8					4									
8						2											2	
9				8	2		4											
10											8	4						2
11										8		8						4
12										4	8							8
13														8	4			
14													8		8			
15													4	8				
16	4	2															2	
17		4	4					2								2		
18										2	4	8						

undiscovered after executing the BFS belongs to a cluster outside of the main cluster (the cluster that includes the first node). The algorithm finds the closest node outside of the main cluster and adds it to the main cluster via a high latency (100 ms)/low bandwidth link (1 MBps). This link simulates a satellite connection with unlimited range [29]. This process is repeated until a fully connected graph is attained. This step is very important because the next process of the NTO implementation, the maximum concurrent multi-commodity flow algorithm requires a fully-connected graph. The output of this process is a predicted topology as illustrated in Table 3.3.

3.7.3 Maximum Concurrent Multi-commodity Flow.. The objective of this process is to compute optimum routes for the MANET given a specific network topology and flow demands. The algorithm computes these routes using a fully polynomial time approximation scheme for the Maximum Concurrent Multi-commodity flow paradigm.

```
BFS(V, E, s)

for each u in V -- {s}      ▷ for each vertex u in V[G] except s.
 do color[u] ← WHITE
     d[u] ← ∞
     π[u] ← NIL
color[s] ← GRAY             ▷ Source vertex discovered
d[s] ← 0                    ▷ initialize
π[s] ← NIL                  ▷ initialize
Q ← ∅                       ▷ Clear queue Q
ENQUEUE(Q, s)
while Q is non-empty
 do u ← DEQUEUE(Q)                       ▷ That is, u = head[Q]
     for each v adjacent to u            ▷ for loop for every
                                          node along with edge.
             do if color[v] ← WHITE      ▷ if color is white
                                          you've never seen
                                          it before
                     then  color[v] ← GRAY
                           d[v] ← d[u] + 1
                           π[v] ← u
                           ENQUEUE(Q, v)
     DEQUEUE(Q)
color[u] ← BLACK
```

Figure 3.3: Breadth-First Search Algorithm – WHITE identifies *undiscovered* state, GRAY identifies *discovered but not fully explored* state and BLACK identifies *fully explored* state [22]

This type of network flow paradigm is used to maximizes the amount of flow traveling from various sources to their corresponding destinations subject to the individual link capacity constraints.

The algorithm computes an approximation using fractional flows, as opposed to the actual solution, in order to minimize the computing resources required. However, the approximation scheme used for this research establishes an error bound that can be set arbitrarily close to the actual solution [7]. The running time of the algorithm increases as the error is minimized. In practicality, an approximation that is within one percent of the actual solution is sufficient to determine optimum MANET routes.

The approximation algorithm works as follows:

- First, it initialize all edge lengths of the topology with $\frac{\delta}{\mu(e)}$. This initial length is very small, and it will be used as a base to exponentially increment the lengths as flows congest edges.

- Next, the algorithm cycles through commodities lengthening edges based on a feasible shortest path routing. Once a path is selected, the edge lengths $l(e)$ associated with this path are increased to avoid over utilization of a particular edge.

- The exit condition of the algorithm is when the sum of all edge lengths ($D(l)$) is greater than one. The maximum error is constrained by the value of ε, the constant used to increment the edge lengths $l(e)$.

The complete pseudocode of the algorithm is illustrated in Figure 3.4.

```
Initialize l(e) = δ/μ(e) ∀ e,  x ≡ 0.
while D(l) < 1
    for j = 1 to k do
        d'_j ← d_j
        while D(l) < 1 and d'_j > 0
            P ← shortest path in P_j using l
            u ← min {d'_j, min_{e∈P} μ(e)}
            d'_j ← d_j − u
            x(P) ← x(P) + u
            ∀ e ∈ P, l(e) ← l(e)(1 + εu/μ(e))
        end while
end while
Return (x;l).
```

Figure 3.4: Max Concurrent Flow Algorithm [7]

3.7.4 Summary. The offline portion of the algorithm utilizes network information available prior to mission execution to compute optimum routes. The output of this phase is preloaded to all participating nodes. This process is conducted during mission planning phase, allowing planners to predict network performance for several "what if" scenarios.

3.8 Online Algorithm To React To Inaccurate NTO Predictions

3.8.1 Introduction. The algorithm implements an online agent that monitors network traffic real-time in order to detect network congestions caused by inaccurate NTO predictions or unaccounted events. This agent is executed at predefined intervals that coincide with Kalman filter prediction rate. For example, if the Kalman filter prediction rate is one second ahead of time, this online algorithm will be executed one time per second. This ensure complete synchronization of the agent throughout mission execution. The main functionality of the agent is implemented in the following five key processes: Generate Queue Predictions, Compute Maximum Concurrent Multi-commodity Flow,Generate and Implement routes, Stop Low Priority Flows and Re-enabled Stopped Flows.

3.8.2 Generate Queue Predictions. The agent uses a Kalman filter to monitor the queue that holds incoming packets. The Kalman filter samples the queue size of the incoming link at predetermined intervals to generate estimates. The size of the interval equals the prediction size. For instance, if we sample the queue every five seconds the Kalman filter generates a prediction of the queue size five seconds into the future.

3.8.3 Maximum Concurrent Multi-commodity Flow. The online agent uses the maximum concurrent multi-commodity flow algorithm previously described, to compute new routes when congestion is detected. Network congestion is detected before queues are overflowed and nodes are forced to drop incoming traffic by monitoring the

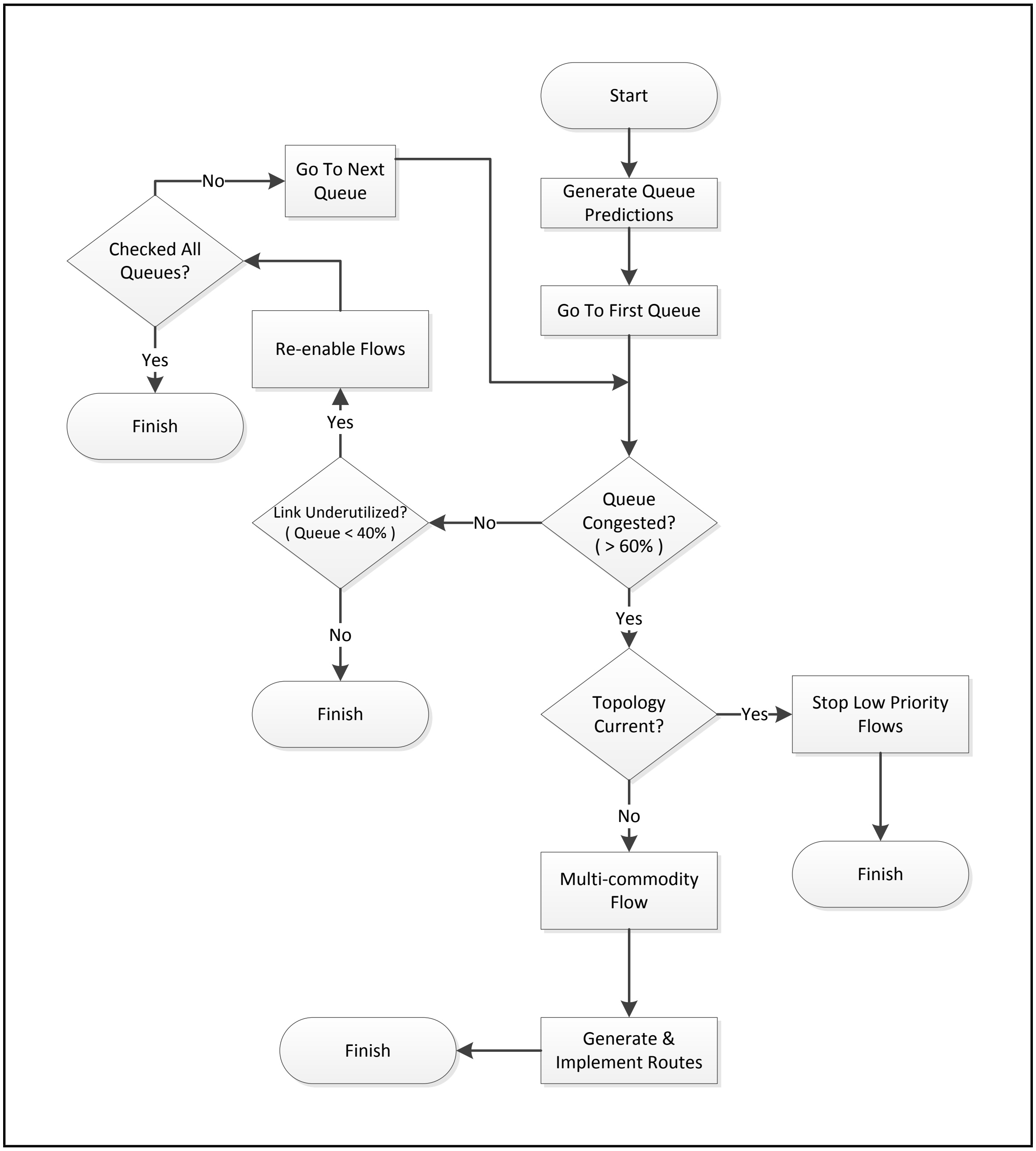

Figure 3.5: Online Agent Flowchart

predictions generated by the Kalman filter. The agent needs to know the entire network topology and network flows before running the algorithm. It is possible to disseminate this information throughout the network within a second in a MANET consisting of twenty active nodes. This algorithm is processor intensive; however, it is possible to compute a solution for a twenty nodes MANET with forty commodities in a fraction of a second using a modern personal computer.

3.8.4 Generate and Implement Routes. The output of the Maximum Concurrent Multi-commodity flow algorithm provides the information necessary to generate optimum routes. The new routes have to be disseminated throughout the entire MANET before they become effective. It is possible to disseminate the new routing information in a MANET consisting of twenty nodes in a fraction of a second. There will be times where the flow demands exceed the capacity of the MANET. However, the routes generated by the multi-commodity flow algorithm will be optimum even under this conditions.

3.8.5 Stop Low Priority Flows. The online agent will detect and edge congestion via Kalman filter predictions. The threshold used to determine edge congestion is a Kalman filter prediction exceeding sixty percent queue capacity. Previous research have demonstrated that Kalman filters can be used effectively to determine a condition where flow demands exceed actual network capacity [12]. A conservative approach to this scenario is to stop the low priority flows associated with this link. This will ensure that high priority traffic is delivered to its destination.

3.8.6 Re-enable Stopped Flows. The algorithm utilizes Kalman filter predictions to determine if a link is saturated. A link is considered underutilized if the prediction for the queue size falls below forty percent. All flows that were stopped to prevent network congestion are reactivated in this process.

3.8.7 Summary. The online agent can install new routes with an optimum solution within three seconds of detecting network congestion. Kalman filter predictions five seconds into the future are necessary to ensure that new routes are installed before nodes are forced to drop incoming traffic. The complete implementation of the online agent is illustrated in Figure 3.5.

3.9 Network Simulator Version 2

All simulations used to support this research were performed using Network Simulator Version 2 (NS2). This simulator is an excellent option to conduct MANET research because the software is very mature. NS2 is a proven discrete event simulator capable of emulating the characteristics of dynamic networks with remarkable results.

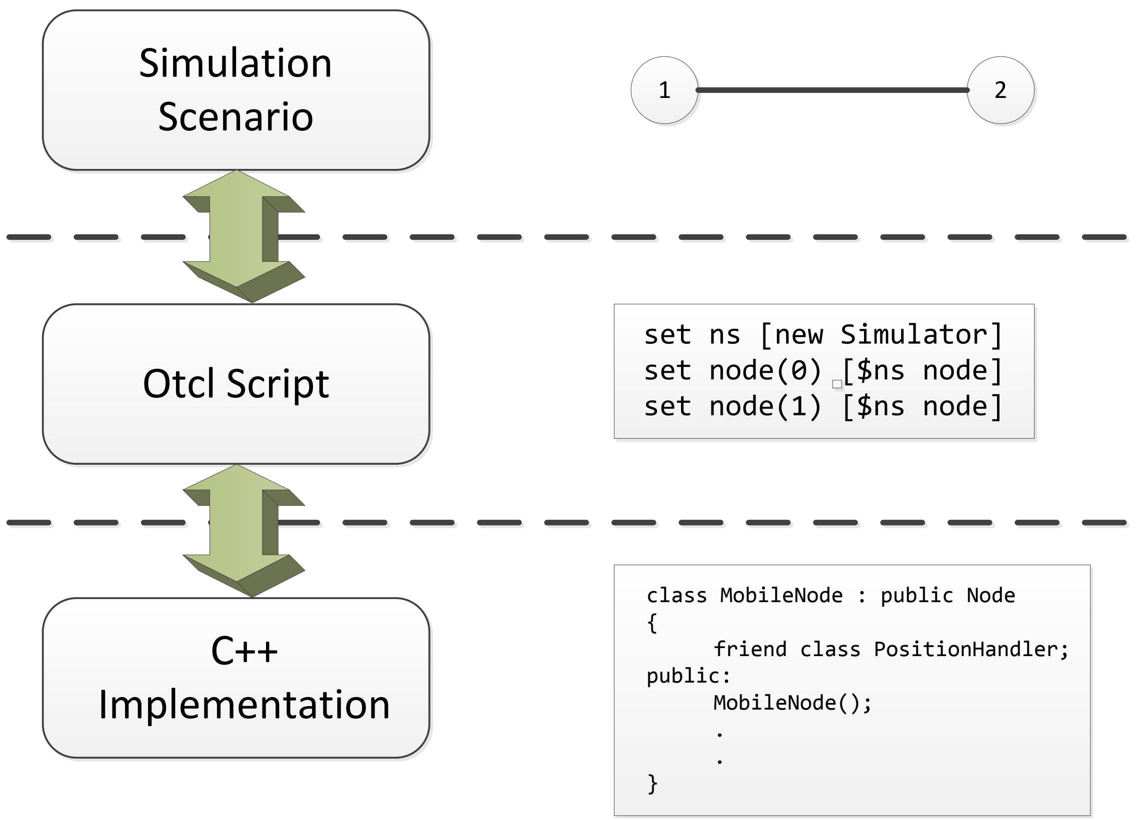

Figure 3.6: NS2 Architecture

3.9.1 Background. NS2 is built using C++ and Object Oriented Tool Command Language (OTcl). This simulator is distributed in source-code format under GNU General Public License enabling users all over the world to join the development of the software.

The foundation for NS2 development, REAL Network Simulator, began in 1989 at the University of California and Cornell University. The software is still actively supported by a growing community with tens of thousands members. Additionally, several government and private institutions have provided substantial contributions to the project including: University of California, Defense Advanced Research Projects Agency, Virtual InterNetwork Testbed and the National Science Foundation. As of the publication of this document, the date for the latest public release of this software is November 4 2011 for version 2.35 [14].

All simulations in NS2 are constructed via OTcl scripts that invoke objects that have been precompiled in C++. NS2 does not include a Graphical User Interface or Integrated Development Environment to generate the simulation code. However, simulation results can be visualized using Network Animator (NAM) software. NAM supports topology layout, packet level animation and various data inspection tools [32].

3.9.2 NS2 Architecture. All NS2 simulations contain two set of programming languages: C++ and OTcl. C++ is used to implement simulation objects because of the speed, and OTcl is used as a frontend setup for the simulator. OTcl allows users to configure objects and schedule events simply and fast. The relationship of this two programming languages is illustrated in Figure 3.6.

3.10 Summary

In conclusion, the algorithm utilizes the NTO concept to develop optimum routes offline prior to mission execution. The algorithm also incorporates an online agent that monitors the current state of the network in order to detect deviations from planned routes and refreshes the topology in real-time as necessary. Additionally, the online agent detects network congestion by monitoring queue sizes. A Kalman filter is utilized to predict network congestion before nodes are forced to drop packets. The online agent sends a stop signal to the low-priority traffic source node when network congestion is detected, in order to minimize any network performance degradation. The online agent uses Kalman filter predictions to determine network edge undertilization. Finally, the online agent sends the restart signal to the respective source nodes for all stopped traffic associated with network edges that are underutilized.

4 Results and Analysis

4.1 Introduction

This chapter describes the experiments performed to demonstrate an implementation of the NTO concept that improves MANET's performance in a military environment. Additional experiments were conducted to demonstrate the usage of Kalman filters to predict network congestion. These predictions show how to establish network controls that minimize any adverse performance impact on the network due to edge overutilization.

- The first set of simulations are performed using an implementation of the AODV routing protocol. These simulations establish a performance baseline that is used to quantify the improvements obtained using the NTO concept as compared to AODV based routing mechanisms.

- The second set of simulations use the NTO concept to compute optimum network routes prior mission execution. The set of routes obtained using this process are preprogrammed into the simulation scenario. The performance of these simulations is compared and contrasted to AODV routing.

- The first and second set simulations utilize a perfect trajectory for all nodes. To simulate deviation from the pre-planned routes, a node's position error was introduced. The first two experiments were reaccomplished taking in consideration this error.

- The final set of simulations utilizes a Kalman filter based online agent to detect deviations from routes computed using the NTO concept and refreshes the network topology accordingly. The online agent is also used to detect network congestion and establishes measures to alleviate congested agents while minimizing adverse impact. The performance of these simulations is compared and contrasted to AODV routing.

4.2 Maximum Concurrent Multi-Commodity Flow Execution Time

The NTO concept is used to predict the network topology at any given time during mission execution. The NTO can also be used to determine the type of network traffic expected during mission execution. The offline algorithm exploits this information in order to maximize network performance by computing optimum routes for these network conditions. An approximation scheme to the Maximum Concurrent Multi-Commodity flow paradigm is used to obtain optimum routes. The online agent also utilizes this approximation scheme to compute network routes on-the-fly when required.

```
vendor_id : GenuineIntel
cpu family : 6
model : 26
model name : Intel(R) Core(TM) i7 CPU  950 @ 3.07GHz
stepping : 5
cpu MHz : 3037.892
cache size : 6144 KB
cpu cores : 8

...

             total       used       free     shared    buffers     cached
Mem:         3528M       934M      2594M         0M        91M       536M
Low:          832M       155M       677M
High:        2695M       778M      1916M
-/+ buffers/cache:       306M      3222M
Swap:        2930M         0M      2930M
```

Figure 4.1: CPU and Memory Profile

This approximation algorithm is processor-intensive, and it can take a considerable amount of time to finish execution if the network topology is very large, or if the maximum error allowed in the approximation is too small. It is necessary to characterize the execution time of this algorithm, because the online agent computes routes real-time during mission execution. The maximum execution time required to execute the algorithm will be used to determine how quickly the online agent is able to load new routes to all participating nodes. The execution time described herein is measured in real time, not simulated within NS2.

The Multi-Commodity flow algorithm was executed one hundred eighty six times in order to profile the execution time. All tests were conducted in a personal computer. The specifications of the personal computer used are shown in Figure 4.1. Network traffic and topologies used to conduct the simulations to support this research were extracted in order to create this execution-time profile. These tests demonstrated that new network routes can be computed in less than 0.30 seconds. This execution time indicates that it is possible to compute network routes using this approach in real-time. The complete set of results for these tests are shown in Table 4.1.

Table 4.1: Maximum Concurrent Multi-Commodity Flow Execution Time

Commodities	**Nodes**	**Executions**	**Total Time (seconds)**	**Time(seconds)**
45	20	31	9.013	0.2907
45	20	31	7.928	0.2557
45	20	31	8.768	0.2828
45	20	31	8.760	0.2826
45	20	31	8.475	0.2734
45	20	31	8.477	0.2735
			Mean	0.2765
			Std Deviation	0.0121

4.3 Network Traffic for Caspian Sea Scenario

The Caspian Sea Scenario simulates a MANET that disseminates information collected from several different ISR sensors for further processing. Nodes one, four and six are used as destination nodes because they simulate Command and Control (C2) airborne platforms. Node twenty-one simulates a remote ground location that is overseeing the entire battlefield. The vast majority of the traffic uses node twenty-one as the destination.

Table 4.2: Network Traffic for Caspian Sea Scenario

Source	Destination	Start (s)	Finish (s)	Packet Size (bytes)	Rate (s)	Priority
0	4	60	120	100	0.001	00
2	1	60	120	100	0.001	01
3	1	60	120	100	0.001	02
4	1	60	120	100	0.001	03
5	21	60	120	250	0.001	04
5	1	60	120	100	0.001	05
6	1	60	120	100	0.001	06
7	1	60	120	100	0.001	07
9	21	60	120	100	0.001	08
10	21	60	120	100	0.001	09
11	21	60	120	100	0.001	10
12	21	60	120	100	0.001	11
13	21	60	120	100	0.001	12
14	21	60	120	100	0.001	13
3	6	60	120	100	0.001	14

All simulations were conducted using the traffic pattern specified in Table 4.2. Nodes acting as traffic sources transmit fifty percent of the time, and are quiet the remaining fifty percent. All traffic sources transmit at a Constant Bit Rate (CBR) when active. Transmitting and quiet times are determined using a Pseudo-Random Number Generator (PRNG). The seed used for the PRNG corresponds to the respective experiment number. For example, experiment eighteen uses a seed of eighteen. This procedure ensures that the same traffic pattern is generated for all different routing solutions.

4.4 Network Performance Using NTO Concept

The first sets of experiments were conducted using AODV routing protocol. The performance of the AODV routing protocol was determined by running the same scenario thirty times in accordance with the Central Limit Theorem. A PRNG was used to determine the transmission intervals. As a result, all traffic sources were transmitting fifty percent of the time on average. The packets sizes and the sending rate remained constant for all transmitting sources.

The second sets of simulations were conducted using the NTO concept to generate optimum routes. All routes used for these experiments were pre-computed using the Maximum Concurrent Multi-Commodity Flow. All other simulation parameters remained unchanged from the first set of experiments.

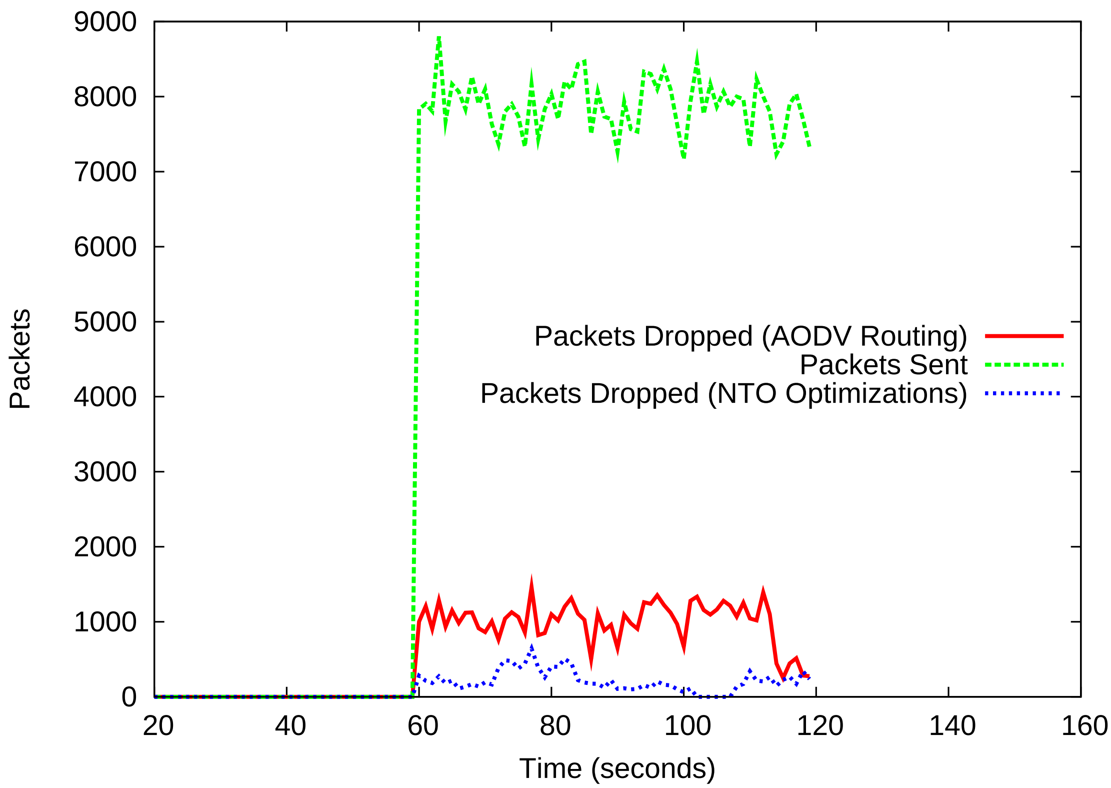

Figure 4.2: Packets Sent vs Packets Dropped Per Second– Average of 30 Simulations

Simulation results demonstrated that network routes obtained using the Multi-Commodity Flow algorithm outperforms traditional routing solutions such as AODV as illustrated in Figure 4.2. Simulations that incorporated NTO optimizations dropped 12,845 packets per second on average. Simulations using AODV as the routing solution dropped 59,841 packets per second on average. It is important to note that the performance of both routing solutions is identical for the last two seconds of simulations due to the fact that there is only one possible route for all active traffic. Consequently, AODV and NTO based solution generate the same routes for the last two seconds.

The throughput for simulations that incorporated NTO optimizations was 843 KBps on average. Simulations based on AODV routing solutions averaged 758 KBps. It is important to note that the performance of both routing solutions is identical for the last two seconds of simulation time because there is only one possible source-to-destination route to deliver the traffic during that timeframe. These results are illustrated in Figure 4.3.

4.5 Deviations from Predicted Routes

Routes generated using the Maximum Multi-commodity Flow algorithm are highly optimized for a given topology. These routes are static and they do not have mechanism to adjust for changes in topology. It is possible to end up with routes that are unfeasible by just changing the quality of one of the MANET's link. One of the most challenging aspects of designing a new MANET routing solution is to incorporate network controls to adjust the routing solution to sudden topology changes.

The first two sets of simulations computed the quality of the network edges based on the position of all participating nodes. Each node's position was computed by interpolating perfect trajectories. Such level of accuracy cannot normally be achieved outside simulation environment.

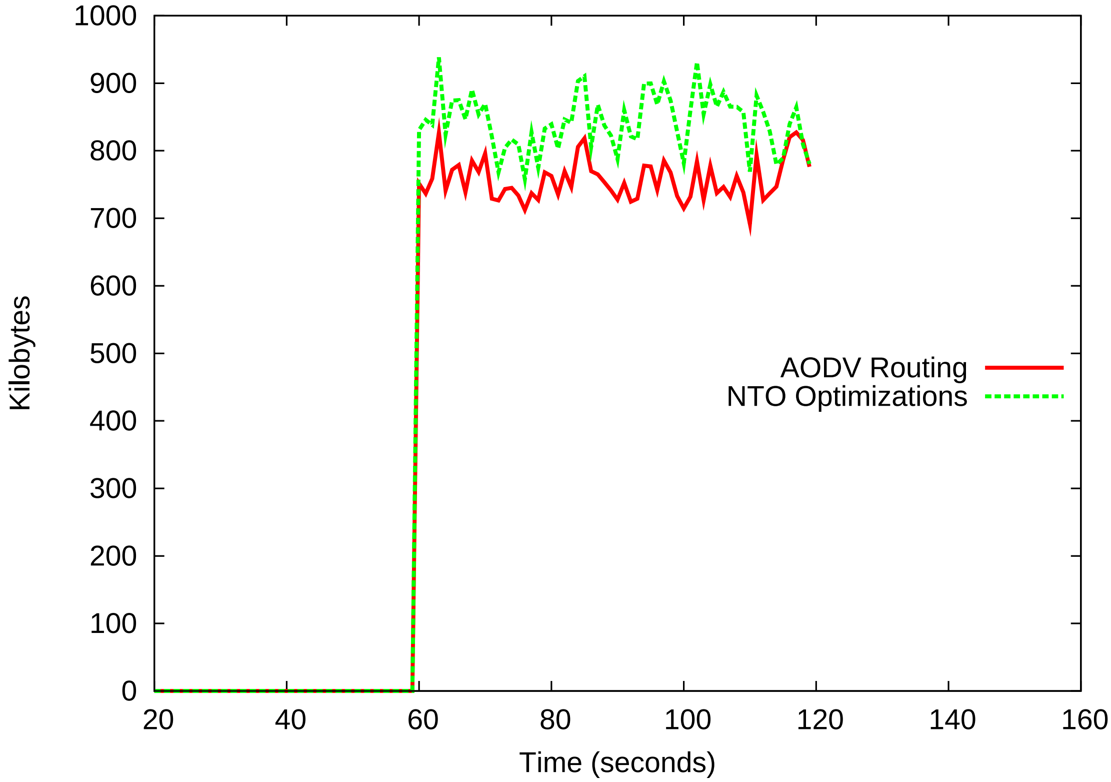

Figure 4.3: Network Throughput – Average of 30 Simulations

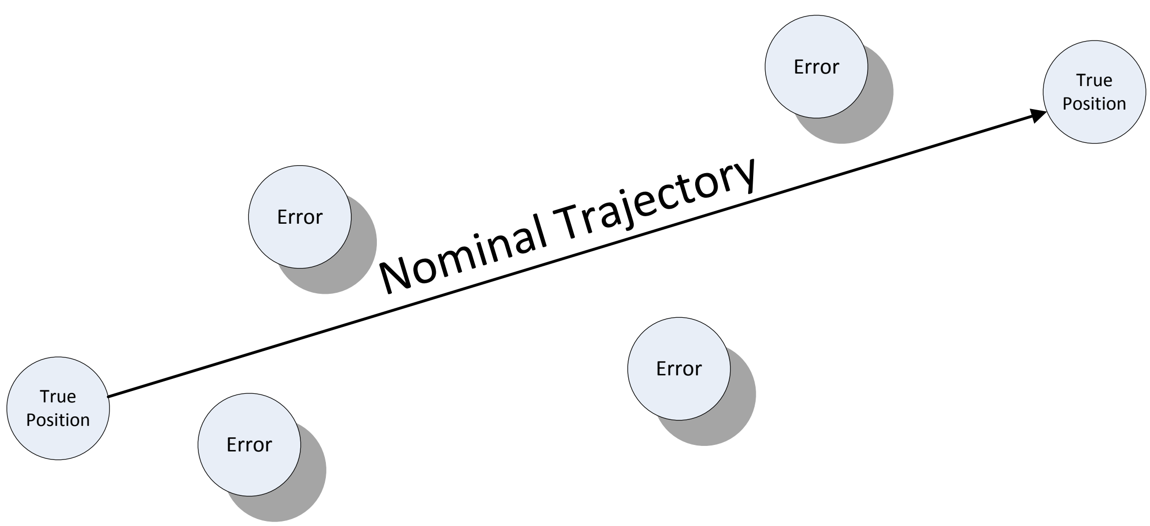

Figure 4.4: Node's Position Error

The third and fourth sets of simulations incorporate an induced error in each node's position. The error is calculated using the following formula:

$$simulated\_node\_position = (actual\_node\_position \cdot ((PRNG() \mod 15) - 7))$$

.

The induced error is noncumulative and will not alter node's overall trajectory as illustrated in Figure 4.4.

4.6 Network Performance Using NTO Concept and Allowing Deviations from Predicted Routes

The next two simulations incorporate the node positioning error model illustrated in Figure 4.4. All other simulations parameters remained unchanged from the first two sets of experiments. This error model simulates deviations from predetermined routes as well as variations on signal strength for radio frequency transmission.

The performance of these two sets of simulation degraded when compared with the first two sets. This change in performance is expected, because the topology changes are more frequent. The optimized routes computed using the Maximum Concurrent Multi-Commodity Flow algorithm were not adjusted to reflect the topology changes induced by the node's position error.

The simulations that incorporated NTO optimizations dropped 15,217 packets per second on average. Simulations that used a routing solution based on AODV protocol dropped 68,104 packets per second on average. It is important to note that the optimized routing solution outperforms the AODV based solution significantly during the last two seconds of simulation time in contrast to the results shown in Figure 4.2. This is the case because AODV saturates few network edges by dynamically adjusting network routes to use shortest-path between source and destination. These results are illustrated in Figure 4.5.

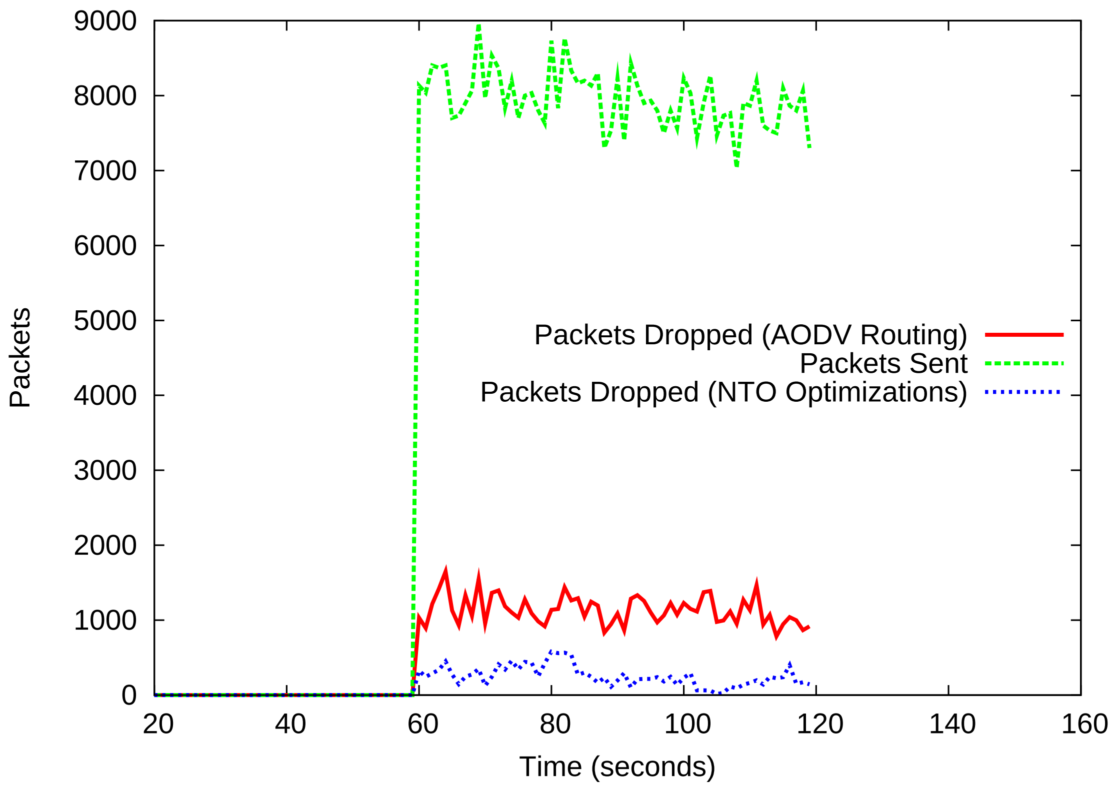

Figure 4.5: Packets Sent vs Packets Dropped Per Second with Topology Errors – Average of 30 Simulations

The mean throughput of the MANET using AODV routing protocol is 751 KBps. The mean network throughput for the routing solution with NTO Optimizations is 843 KBps on average. These results are illustrated in Figure 4.6.

The induced position error did not affect the throughput performance significantly for either routing solution. The position error reduced the performance of AODV routing protocol by 0.91%. The position error reduced the performance of the optimized routing solution by 0.47%. AODV routing protocol is more susceptible to rapid topology changes because it adjust routes dynamically for every topology change.

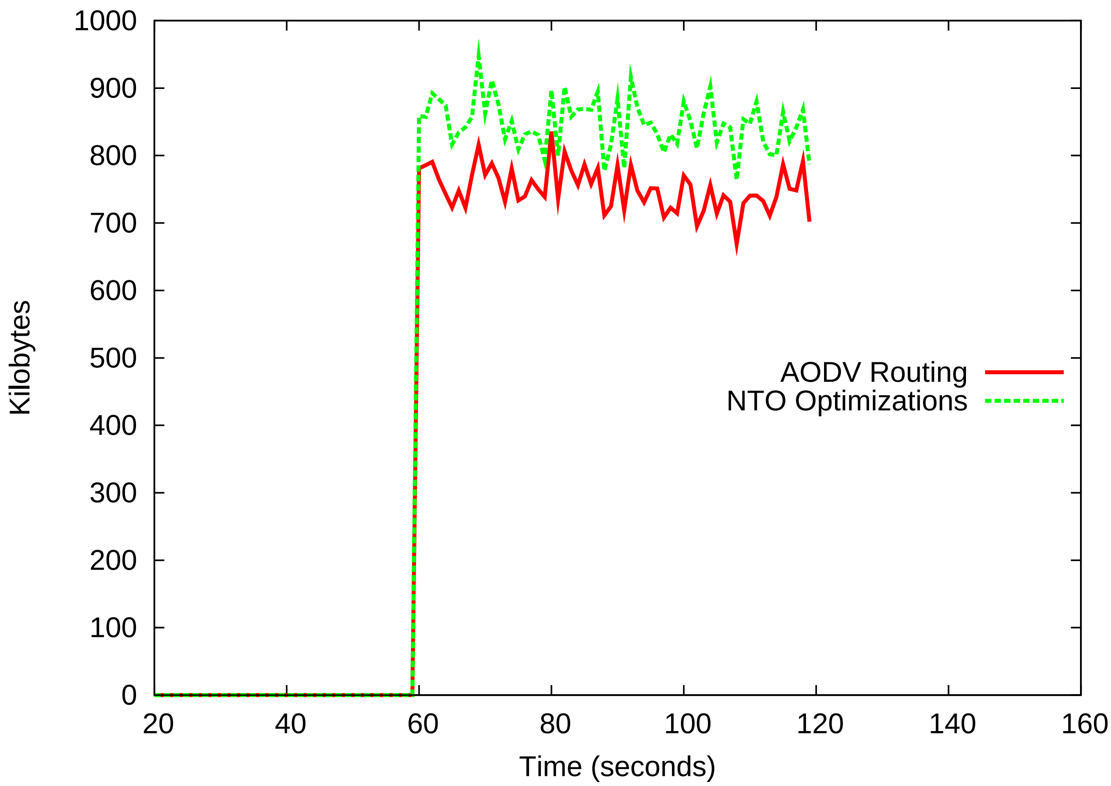

Figure 4.6: Network Throughput with Topology Errors – Average of 30 Simulations

4.7 Confidence Interval Comparison with and without Topology Errors

4.7.1 Network Throughput. Visual inspection of Figure 4.7 demonstrate that the 95% confidence interval overlaps for network throughput comparison. This fact can be verified mathematically using the following method: If the means' difference is less than two times the standard error, as shown in Equation 4.1, then the two 95% confidence intervals overlap.

$$mean_b - mean_a < 2(SE_a + SE_b) \qquad (4.1)$$

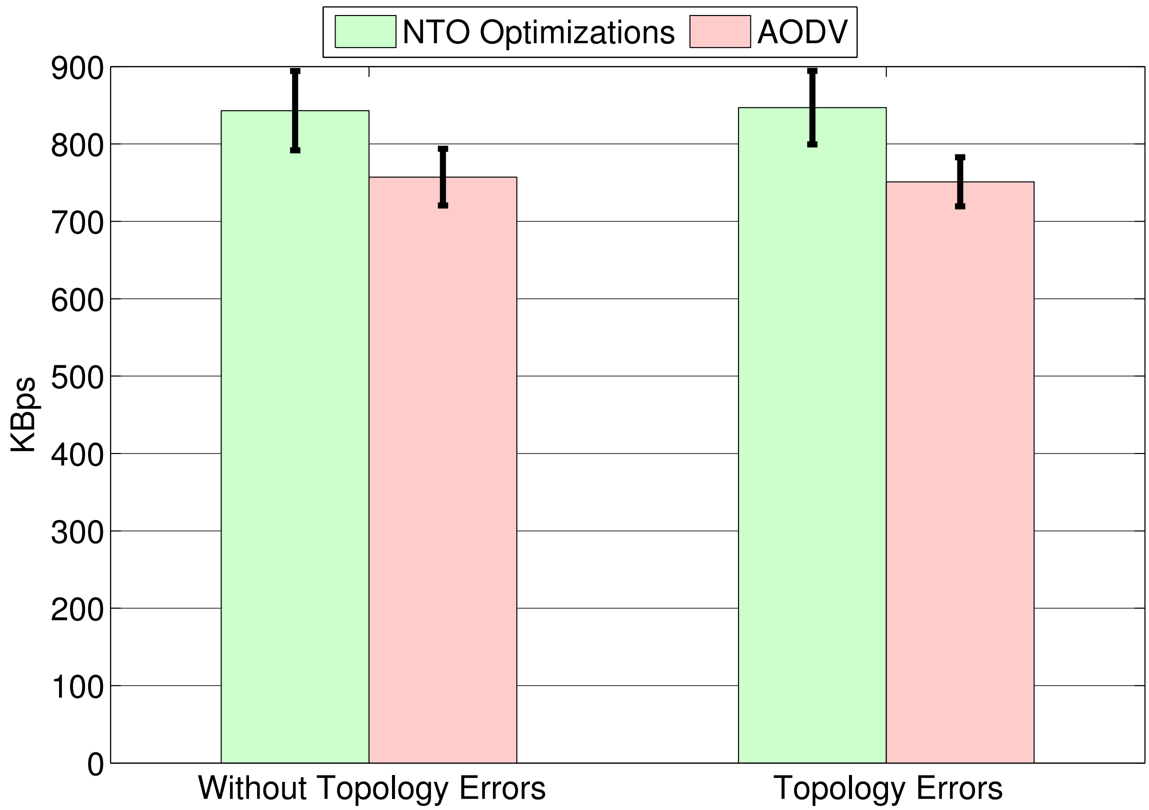

Figure 4.7: Network Throughput – 95% Confidence Interval Comparison with and without Topology Errors

A mathematical comparison of the 95% confidence interval for the network throughput without topology errors shows that the confidence intervals overlap.

$$(847.30 - 757.71) > 2(23.76 + 18.31) \Rightarrow (89.59 > 84.14) \tag{4.2}$$

A mathematical comparison of the 95% confidence interval for the network throughput with topology errors shows that this confidence intervals overlap as well.

$$(843.34 - 750.88) > 2(25.62 - 15.88) \Rightarrow (92.46 > 83.00) \tag{4.3}$$

However, it is possible to have two means that are statistically significantly different even though the confidence intervals overlap. The following rule can be used to determine if two means are statistically different from each other at the 0.05 level [37]:

$$mean_b - mean_a > 2\sqrt{SE_a^2 + SE_b^2} \tag{4.4}$$

Using this rule we can determine if there is any statistical difference between routing NTO Optimizations versus AODV.

- Without Topology Errors
  - $(843.34 - 750.88) > 2\sqrt{25.62^2 + 15.88^2}$
  - $92.46 > 60.28$
- With Topology Errors
  - $(847.30 - 757.71) > 2\sqrt{23.76^2 + 18.31^2}$
  - $90.59 > 59.99$

The two means are statistically significant difference at the 0.05 level. This difference highlights the fact that routing solutions that implement NTO optimizations attain higher network throughput than AODV routing protocol.

4.7.2 Dropped Packets. Visual inspection of Figure 4.8 shows that the confidence intervals do not overlap, demonstrating that the two means are statistically different at the 95% confidence level.

Routing solutions that implement NTO optimization minimize dropped packets because it uses a Maximum Concurrent Multi-Commodity Flow paradigm to model the network traffic and topology. This technique minimizes edge overutilization. In contrast, AODV routing protocol uses the minimum cost path between any given source-destination pair. This routing solution saturates network edges that are part of the minimum cost path for multiple source-destination pairs.

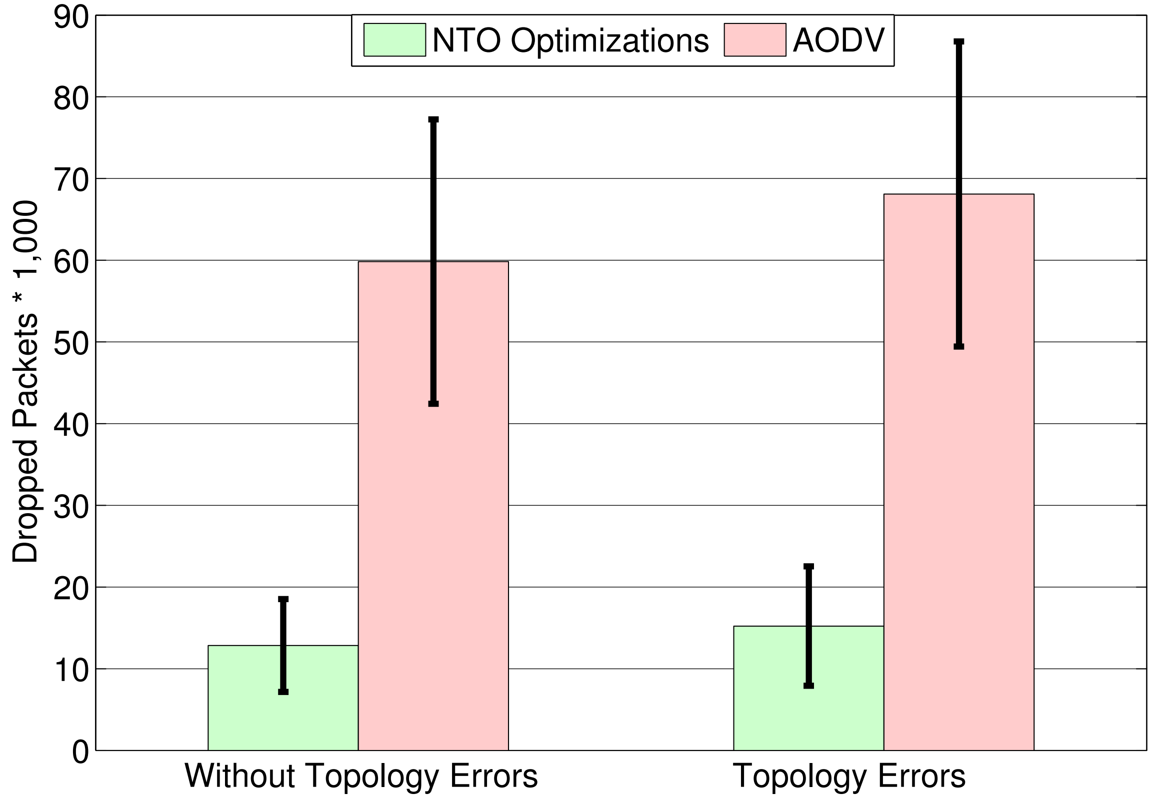

Figure 4.8: Dropped Packets – 95% Confidence Interval Comparison with and without Topology Errors

4.8 Network Performance Using NTO Concept, Kalman Filters and Allowing Deviations from Predicted Routes

4.8.1 Kalman Filter Tuning. The maximum throughput that can be sustained in a MANET is very diverse due to the wide variety of possible network configurations and operating environment. Because of these conditions, it is necessary to tune the Kalman filter to the MANET environment before it can provide adequate estimates of the queue size [30]. Improperly tuned filters can show slow frequency response or oscillations making them unsuitable to establish network controls.

The Kalman filter was tuned to match the parameters of the simulation environment by modifying the weights of its covariance dynamic noise vector iteratively, until the desired performance was achieved. The response of the tuned filter can be used to accurately estimate queue size given noise corrupted measurements as illustrated in Figure 4.9.

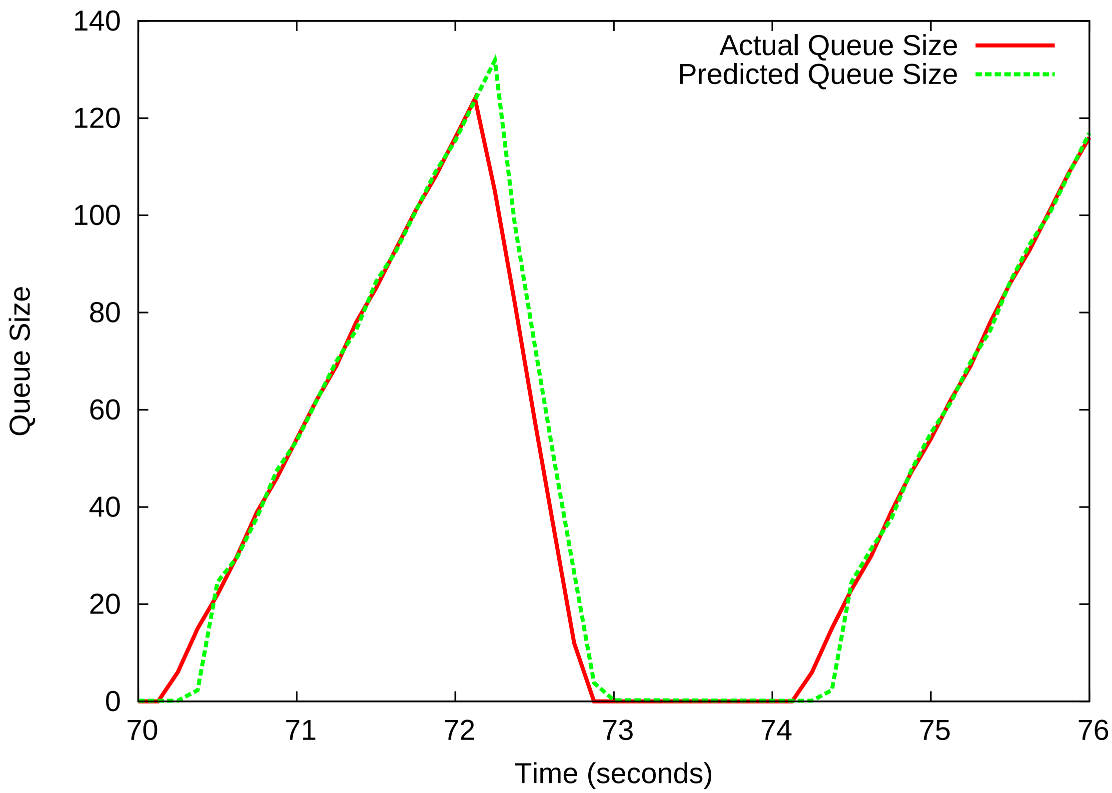

Figure 4.9: Kalman Filter Tuning

4.8.2 Traffic Regulation Using Kalman Filters. This research uses Kalman filter to regulate incoming traffic in order to minimize packet loss. The Kalman filters are installed in the nodes, where they monitor all incoming packets. The Kalman filter predicts node's queue size a second, or a fraction of a second, into the future based on several actual queue size measurements. Nodes using Kalman filters stops low priority traffic when queue size prediction exceeds 60% maximum queue size (150 packets) preventing edge congestion. All nodes are configured with a drop-tail queue, with a maximum queue capacity of 250 packets. Nodes using Kalman filters restart all stopped flows once the Kalman filter prediction is below 40% maximum queue size (100 packets).

Using this technique, it is possible to maximize network utilization, while minimizing packets drops. Figure 4.11 illustrate how Kalman filters can be used to regulate traffic in order to maximize bandwidth utilization without congesting the network edges. During this simulation low priority packets were regulated to prevent queue saturation.

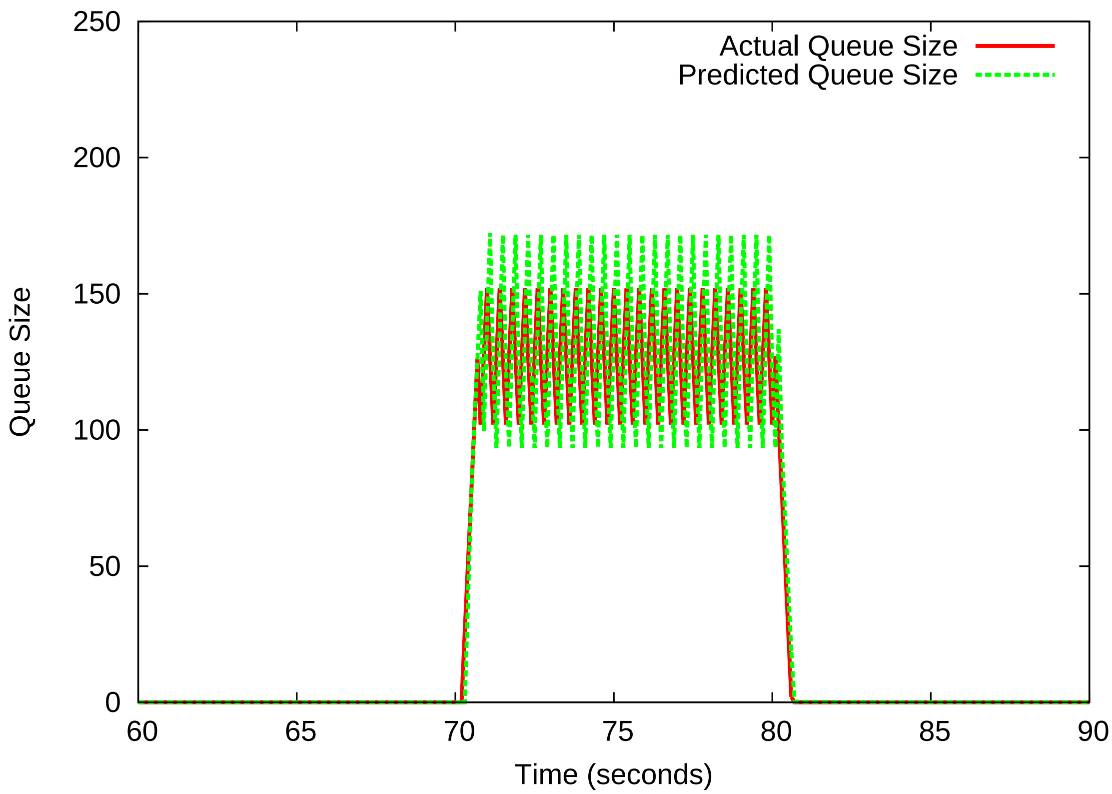

Figure 4.10: Kalman Filter Regulating Traffic – 1/8 Second Predictions

4.8.3 Network Performance Using Kalman Filters . The final set of simulations utilizes a Kalman filter based online agent to detect deviations from routes computed using the NTO concept and refreshes the network topology accordingly. Nodes detect deviations from NTO preplanned routes whenever the plan requires forwarding traffic via non-existing network edges or when edge saturations occur. Detection of network edge saturation can take up to one second under this condition because all nodes in the network path use queues to buffer excess traffic.

Network topology is refreshed using traditional node discovery methods. One hundred and eighty experiments were conducted to determine the time it takes to replicate network topology for a MANET with twenty nodes. In all cases the MANET topology was obtained in less than 0.30 seconds after transmitting less than 300 packets. Table 4.3 shows the total time required to refresh network routes to all participating node.

Table 4.3: Time required to refresh network routes

Detect deviation from preplanned routes	1 second
Refresh network topology	0.3 seconds
Compute optimum network routes	0.3 seconds
Disseminate routes to participating nodes	0.3 seconds
Total time	**~ 2 seconds**

The preplanned topology loaded for this set of simulations was offset by 100 minutes to simulate deviations from preplanned routes. This will ensure that network routes preloaded to nodes are not consistent with the actual topology. The online agent detects this failed state and implements network controls to recover.

The online agent detects deviations from NTO preplanned routes in the form of non-existing network edges, saturated edges, or both and computes optimum network routes in real-time. Results indicate that the online agent generates routes that provide better throughput performance than AODV routing protocol. Additionally, the online agent stops low priority traffic when network demands exceed actual capacity minimizing high priority traffic loss. The results are illustrated in Figure 4.11.

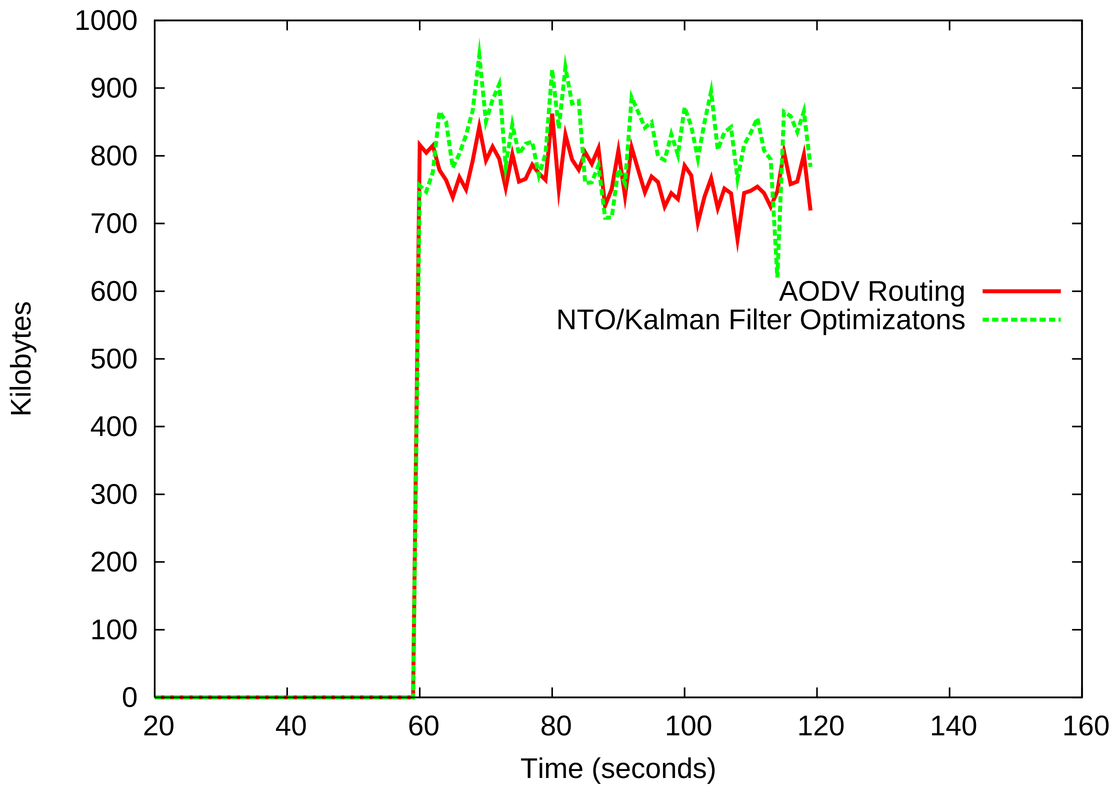

Figure 4.11: Network Performance Using NTO Concept, Kalman Filters and Allowing Deviations from Predicted Routes

4.9 Summary

This chapter presented the analysis and results for the simulations and experiments run during the course of this research. Simulations results demonstrate that the overall network throughput increases when the NTO concept is applied compared to AODV routing protocol. Additionally, simulations results indicated that network routes computed using Maximum Concurrent Multi-Commodity flow algorithm are not susceptible to rapid topology variations induced by noise.

5 Conclusions

Military environment require highly dynamic MANETs to meet operational mission requirements. Military commanders rely on the timely delivery of critical battlefield information to make decisions quickly and as accurately as possible. However, traditional MANET routing protocols do not provide quality of service. Furthermore, they do not implement active controls to minimize the impact of network congestion. Using the NTO concept it is possible to optimize network routes to minimize edge overutilization and increase network throughput.

5.1 Research Impact

Using the NTO concept, this research demonstrated that it is possible to optimize the performance of the highly dynamic MANET used to deliver critical battlefield information to the warfighter. The NTO process implements network controls to provide quality of service in highly dynamic network environments. The increased information flow can be used to achieve information superiority.

A very controlled environment was used to perform the required simulations to support this research. The environment was controlled in order to measure the impact on network throughput and packet loss on MANETS when the NTO concept is incorporated in the routing solution.

Even thought though the results demonstrate better performance, this routing solution cannot be used for every scenario. Consideration must be given to the additional resources required to implement and manage this solution. Nodes are required to have significant computer power in order to implement the Maximum Concurrent Multi-Commodity Flow algorithm.

5.2 Contributions

As previously identified, the goals of this research were to develop an algorithm that capitalizes on the network information implied in the Air Tasking Order. The algorithm must satisfy the following criterion:

- Compute optimum routes based on predicted network traffic and topology.
- Implement agents monitoring network traffic real-time to detect and react to inaccurate network predictions minimizing any adverse impact.
- Minimize packet loss when compared to AODV routing protocol.
- Maximize network throughput when compared to AODV routing protocol.

Several simulations were performed to demonstrate the algorithm's effectiveness. It was shown that the algorithm performs at least as well as AODV when mission execution significantly deviates from pre-planned routes (worst case scenario). When mission execution follows preplanned routes, a 12% improvement in mean network throughput was achieved as illustrated in Figure 4.6. Additionally, results indicate a 10 % decrease in the number of packets dropped per second as illustrated in Figure 4.5.

5.3 Future Work

Although results contained in this research are very promising, there is much work that can be done to further refine the methods specified in this document. Specifically, future work should include:

- Additional scenarios: gather real-world data and compare it to planning documents (e.g. ATO) and quantify the differences.
- Compare the performance of this algorithm with other MANET protocols such as Optimized Link State Routing or Dynamic Source Routing.

- Forecast network topology considering nodes' antenna radiation and reception patterns.

5.4 Summary

This research demonstrates that it is possible to generate optimum network routes whenever the following information can be predetermined:

- Network topology. The network topology must consist of a graph with viable path for all source-destination pairs for a specified period of time.

- Data transmission requirements. The data rate and source-destination information for the vast majority of data flows (90% or more) must be known for a specified period of time.

These requirements fit very well ISR missions where wireless sensors are constantly transmitting measurements, and airborne platforms move in predetermine patterns. It is possible to derive these requirements from ATOs for more complex military missions. Simulations demonstrated that routing solutions implementing NTO optimizations minimizes packet loss and consequently increases network throughput.

Bibliography

[1] Chakeres, I.D. and L. Klein-Berndt. "AODVjr, AODV Simplified". *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(3):100–101, 2002.

[2] CNSS. *National Policy Governing the Use of High Assurance Internet Protocol Encryptor (HAIPE) Product*. Technical report, Committee on National Security Systems, 2007. URL http://www.cnss.gov/Assets/pdf/CNSSP-19.pdf.

[3] Compton, M., K. Hopkinson, and S. Graham. "The Network Tasking Order (NTO)". *IEEE, Military Communications Conference*, 2008.

[4] Das, S., C. Perkins, and E. Royer. "Ad Hoc On Demand Distance Vector (AODV) Routing". *Mobile Ad-hoc Network (MANET) Working Group, IETF*, 2002.

[5] Du Plessis, R.M. "Poor Man's Explanation of Kalman Filtering". *Autonectics Division of North American Rockwell Corporation*, 1967.

[6] Even, S., A. Itai, and A. Shamir. "On the Complexity of Time Table and Multi-Commodity Flow Problems". *Foundations of Computer Science, 1975., 16th Annual Symposium on*, 184 –193. Oct. 1975. ISSN 0272-5428.

[7] Fleischer, L.K. "Approximating Fractional Multicommodity Flow Independent of the Number of Commodities". *SIAM Journal on Discrete Mathematics*, 13:505, 2000.

[8] Giordano, S. "Mobile Ad Hoc Networks". *Handbook of Wireless Networks and Mobile Computing*, 23, 2001.

[9] Gocmen, M., K. Hopkinson, and M. Compton. "The Benefits of a Network Tasking Order in Combat Search and Rescue Missions". *IEEE, Military Communications Conference*, 2009.

[10] Goldberg, A., J. Oldham, S. Plotkin, and C. Stein. "An implementation of a combinatorial approximation algorithm for minimum-cost multicommodity flow". *Integer Programming and Combinatorial Optimization*, 1998.

[11] Gonzalez, T.F. *Handbook of Approximation Algorithms and Metaheurististics*, Volume 10. Chapman & Hall, 2007.

[12] Haught, J. *Adaptive Quality of Service Engine with Dynamic Queue Control*. Master's Thesis, Air Force Institute of Technology, 2011.

[13] Holland, Gavin, Nitin Vaidya, and Paramvir Bahl. "A Rate-Adaptive Mac Protocol For Multi-Hop Wireless Networks". *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, 236–251. ACM, New York, NY, USA, 2001.

[14] Issariyakul, T. and E. Hossain. *Introduction To Network Simulator NS2*. Springer Verlag, 2008.

[15] JCS. *Joint Publication 3-30. Command and Control for Joint Air Operations*. Technical report, DTIC Document, 2010.

[16] Junhai, L., Y. Danxia, X. Liu, and F. Mingyu. "A Survey of Multicast Routing Protocols for Mobile Ad-Hoc Networks". *Communications Surveys & Tutorials, IEEE*, 11(1):78–91, 2009.

[17] Kapoor, C. and G. Sharma. "To Improve the QoS in Manets Through Analysis Between Reactive and Proactive Routing Protocols". *Computer Science & Engineering: An International Journal (CSEIJ)*, 2011.

[18] Karimi, M. and Deng Pan. "Challenges for Quality of Service (QoS) in Mobile Ad-hoc Networks (MANETs)". *Wireless and Microwave Technology Conference (WAMICON)*, 2009.

[19] Lee, S.D. *Routing UAVs to Co-Optimize Mission Effectiveness and Network Performance with Dynamic Programming*. Master's Thesis, Air Force Institute of Technology, 2011.

[20] Lefebvre, T., H. Bruyninckx, and J. De Schutter. "Kalman Filters For Non-Linear Systems: A Comparison of Performance". *International journal of Control*, 77(7):639–653, 2004.

[21] Mirhakkak, M. and P. Ta. "Modeling and Simulation of GIG Networking Scenarios Using Efficient HAIPE Discovery". *IEEE, Military Communications Conference (MILCOM)*, 2007.

[22] Muhamma, R. *Breadth-First Search Traversal Algorithm*. Technical report, Kent State University, 2012. URL http://www.personal.kent.edu/~rmuhamma/Algorithms/algorithm.html.

[23] Murray, M.W. *Automatically Generating a Distributed 3D Battlespace Using USMTF and XML-MTF Air Tasking Order, Extensible Markup Language (XML) and Virtual Reality Modeling Language (VRML)*. Technical report, DTIC Document, 2000.

[24] Naveen, K. and K. Karuppanan. "Mobile cluster assisted routing for urban VANET". *International Conference on Recent Trends in Information Technology (ICRTIT)*, 296 –300, June 2011.

[25] NSA. *Global Information Grid*. Technical report, National Security Agency, 2011. URL http://www.nsa.gov/ia/programs/global_industry_grid/index.shtml.

[26] Sang, Aimin and San qi Li. “A predictability analysis of network traffic”. *INFOCOM 2000. IEEE Nineteenth Annual Joint Conference of the Computer and Communications Societies. Proceedings.* 2000.

[27] Scheideler, Christian. *Network Flows III – Multicommodity Flows*. Technical report, Johns Hopkins University, 2003. URL http://www.cs.jhu.edu/~scheideler/courses/600.348_F03/.

[28] Singpurwalla, N.D. “Network Routing in a Dynamic Environment”. *The Annals of Applied Statistics*, 5(2B):1407–1424, 2011.

[29] Stookey, D.E. *A Notional Battlespace For Simulating and Testing Dynamic Wireless Networks*. Technical report, Air Force Institute of Technology, 2006.

[30] Stuckey, N.C. *Stochastic Estimation and Control of Queues within a Computer Network*. Master’s Thesis, Air Force Institute of Technology, 2007.

[31] Subbarao, Madhavi. *Performance of Routing Protocols for Mobile Ad-Hoc Networks*. Technical report, National Institute of Standards and Technology. URL http://w3.antd.nist.gov/wctg/manet/docs/perf_routing_protocols.pdf.

[32] VINT. *Nam: Network Animator*. Technical report, The VINT Project, 2012. URL http://www.isi.edu/nsnam/nam/.

[33] Wang, Z. and J. Crowcroft. “Analysis of Shortest-Path Routing Algorithms in a Dynamic Network Environment”. *ACM SIGCOMM Computer Communication Review*, 22(2):71, 1992.

[34] Welch, G. and G. Bishop. “An Introduction to the Kalman Filter”. *Design*, 7(1):1–16, 2006.

[35] White, O. *Network Centric Operations*. Technical report, Defence Research and Development Canada, 2005.

[36] Willis, J.B. and M.J. Davis. “Design of the Reconnaissance, Surveillance, and Target Acquisition Squadron for the US Army’s Interim Brigade Combat Team”. *IEEE International Conference on Systems, Man, and Cybernetics*, 2000.

[37] Wolfe, R. and J. Hanley. “If We’Re So Different, Why Do We Keep Overlapping? When 1 Plus 1 Doesn’T Make 2”. *Canadian Medical Association Journal*, 166(1):65, 2002.

Vita

Captain Joan Addison Betances is a student at the Air Force Institute of Technology pursuing a Masters Degree in Computer Engineering. He graduated from Antillean Adventist University in Mayaguez, Puerto Rico with the academic degree of Bachelor of Science in Computer Science in 1996. He also obtained a degree in Electrical Engineering from Walla Walla University in 2003. Captain Betances is a member of Tau Beta Pi and Eta Kappa Nu honor societies.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
22-03-2012	Master's Thesis	Aug 2010 - Mar 2012

4. TITLE AND SUBTITLE
Context Aware Routing Management Architecture for Airborne Networks

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Betances, Joan A, Captain, USAF

5d. PROJECT NUMBER
126292P

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GCE/ENG/12-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
Technical Advisor Dr. Robert Bonneau
875 N Randolph St, Ste 325 Rm 3112
Arlington, VA 2203
robert.bonneau@afosr.af.mil - (703) 426-9545

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT
This thesis advocates the use of Kalman filters in conjunction with network topology information derived from the Air Tasking Order (ATO) during the planning phase for military missions. This approach is the basis for an algorithm that implements network controls that optimize network performance for Mobile Ad hoc Networks (MANET). The trajectories of relevant nodes (airborne platforms) participating in the MANET can be forecasted by parsing key information contained in the ATO. This information is used to develop optimum network routes that can significantly improve MANET performance. Improved MANET performance in the battlefield enables decision makers to access information from several sensors that can summarize mission execution status real-time. In one simulated test case there was a 25% percent improvement of network throughput, and 23% reduction on dropped packets. Using this technique we can selectively preserve the Quality of Service (QoS) by establishing network controls that drops low priority packets when necessary. The algorithm improves the overall MANET throughput while minimizing the packets dropped due to network congestion.

15. SUBJECT TERMS
^Air Tasking Order, Network Tasking Order, Kalman Filter, Mobile Ad Hoc Network, Maximum Concurrent Multi-Commodity Flow

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Kenneth Hopkinson, PHD, ENG
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	80	(937) 255-6565;email:kenneth.hopkinson@afit.edu